UCRL-CONF-222359

LAWRENCE LIVERMORE NATIONAL LABORATORY

Seismic interpretation using Support Vector Machines implemented on Graphics Processing Units

H. Anderson Kuzma, J. W. Rector, D. Bremer

June 23, 2006

The IAMG 2006 Annual Conference on Quantitative Geology from Multiple Sources
Liege, Belgium
September 3, 2006 through September 8, 2006

Seismic interpretation using Support Vector Machines implemented on Graphics Processing Units

Heidi Anderson Kuzma[1], James W. Rector[1], Dave Bremer[2]
1 Dept. of Civil and Environmental Engineering, University of California, Berkeley
2 Lawrence Livermore National Labs,
Corresponding author: hkuzma@berkeley.edu

ABSTRACT : Support Vector Machines (SVMs) estimate lithologic properties of rock formations from seismic data by interpolating between known models using synthetically generated model/data pairs. SVMs are related to kriging and radial basis function neural networks. In our study, we train an SVM to approximate an inverse to the Zoeppritz equations. Training models are sampled from distributions constructed from well-log statistics. Training data is computed via a physically realistic forward modelling algorithm. In our experiments, each training data vector is a set of seismic traces similar to a 2-d image. The SVM returns a model given by a weighted comparison of the new data to each training data vector. The method of comparison is given by a kernel function which implicitly transforms data into a high-dimensional feature space and performs a dot-product. The feature space of a Gaussian kernel is made up of sines and cosines and so is appropriate for band-limited seismic problems. Training an SVM involves estimating a set of weights from the training model/data pairs. It is designed to be an easy problem; at worst it is a quadratic programming problem on the order of the size of the training set. By implementing the slowest part of our SVM algorithm on a graphics processing unit (GPU), we improve the speed of the algorithm by two orders of magnitude. Our SVM/GPU combination achieves results that are similar to those of conventional iterative inversion in fractions of the time.

KEYWORDS: SVM, inversion, GPU, AVO

1. Introduction

Seismic amplitude variation with offset (AVO) inversion can be used to recover lithologic properties of potential hydrocarbon reservoirs (Castagna and Backus, 1993). Using the non-linear Zoeppritz equations (Aki and Richards, 1980), the amplitude of a reflected wave can be computed as a function of its angle of incidence and contrasts in pressure wave velocity, shear wave velocity and density across a geologic interface. A Support Vector Machine (SVM) can be trained to approximate a local inverse to the Zoeppritz equations, making it possible to interpret AVO data in a fraction of the time it takes a conventional non-linear iterative inversion to do the same. Further speed gains are made by implementing part of the process on a graphics processing unit (GPU). Graphics cards have up to 350 Gigaflops of potential computing power for algorithms that work within their constraints, and generally cost less than $500.

Although SVMs are directly related to kriging, they are relatively new to the geophysical literature (Vapnik, 2008). They have been proposed for AVO inversion (Kuzma, 2003; Kuzma and Rector, 2004; Li and Castagne, 2003), and facies delineation. (Wohlberg et al., 2006). (Hidalgo et al, 2003) used a related method to invert electromagnetic data. SVMs are implemented on GPUs in (Ohmer et. al, 2005). The type of SVM that is illustrated in this paper (least squares SVM with a Gaussian kernel) is related to a radial basis function neural network (Haykin, 1999).

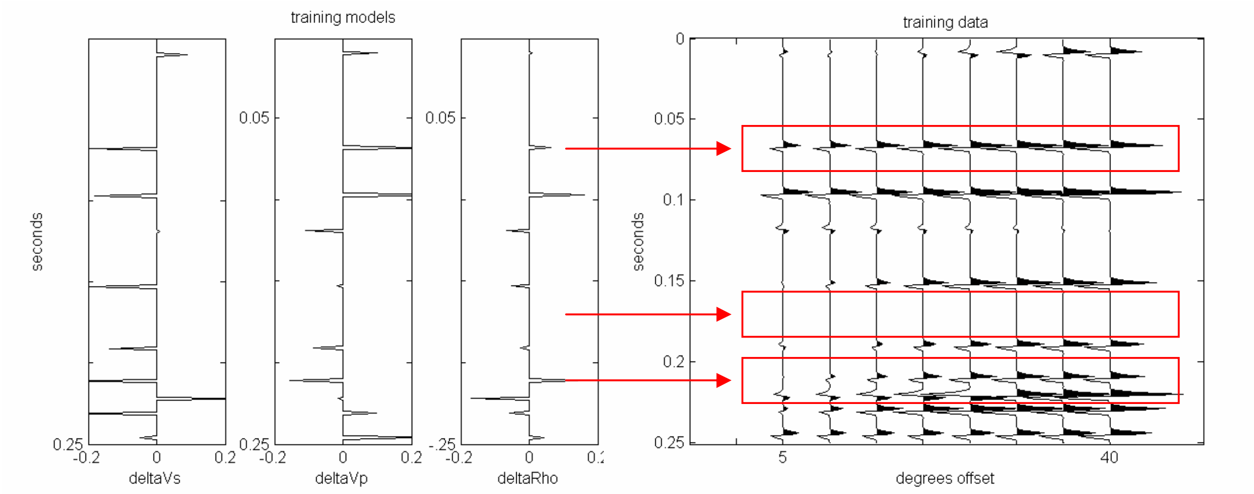

Figure 1 a): Example of training models and training data. Models are sampled using well-log statistics. Data is computed using Zoeppritz equations and convolved with a wavelet. One time sample of the training models corresponds to an image of 11 time samples of training data.

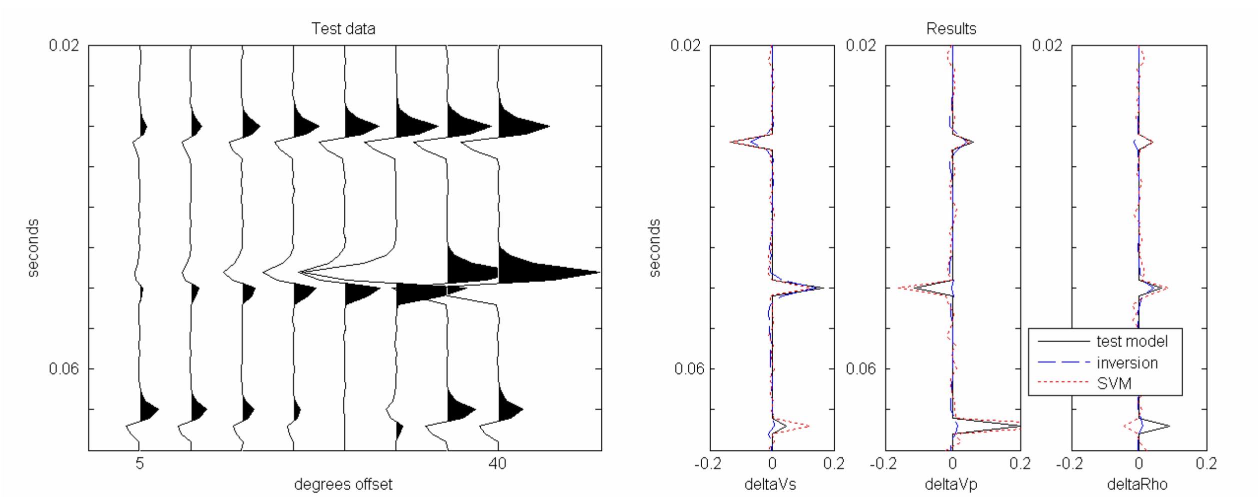

Figure 1 b): Results using inversion vs. SVM to interpret synthetic data with 5% noise.

Method	Model recovery RMS error	Data prediction RMS error	Number of forward models required to invert/train	Time (seconds)
Inversion	0.0165	0.0409	30100	1861
SVM	0.0158	0.0592	918	8.5

Table 1: SVM does a better job finding known models than inversion, but inversion does a better job predicting data. The SVM is much faster than inversion, partly because it can be trained on many fewer models than are necessary to compute the gradients of an iterative inversion.

Method

1.1. Training SVMs to find an inverse relationship

Given a data vector **d**, the goal of inversion is to recover a model vector **m** containing earth parameters of interest. In this study, **m** is made up of reflectivity contrasts in pressure wave velocity, shear wave velocity and density for a series of layers: **m** = [ΔVp/Vp, ΔVs/Vs, Δρ/ρ)]. The elements of **d** are the amplitudes of reflections at offsets from 0 to 40 degrees. Since **d** can be computed from **m** using the Zoeppritz equations, the goal of an SVM for AVO is to find an approximate inverse to the Zoeppritz equations:

$$\mathbf{m} = SVM(\mathbf{d}) \approx Z^{-1}(\mathbf{d}) \quad . \tag{1}$$

Training an SVM requires assembling a training set of known model/data pairs, picking a *kernel* function K and solving for a set of coefficients **α** such that:

$$\mathbf{m_j} = \sum_{i=1}^{n} \alpha_i K(\mathbf{d_i}, \mathbf{d_j}) \quad . \tag{2}$$

The model returned by an SVM is a weighted comparison of a data vector to the training data vectors. The kernel gives the method of comparison.

In our experiments, training models are chosen at random using parameter statistics from well-logs. The choice of training models acts like a Bayesian prior used to regularize an inversion (Kuzma and Rector, 2004). Training data is made using the Zoeppritz equations. The models and data are zero-padded to make "layers" of random thickness and the data is convolved with a realistic wavelet[1]. A subset of training data is shown in Figure 1a). Each data vector includes 11 time samples of data and can be thought of as a 2-dimentional image. We use roughly 1000 training model/data pairs representing about 100 examples of lithologic layers.

Any symmetric positive definite function can be used as a kernel. By Mercer's theorem, a kernel performs a dot product in a *feature space* with a transformation defined by ϕ:

$$K(\mathbf{d_i}, \mathbf{d_j}) = \phi(\mathbf{d_i}) \cdot \phi(\mathbf{d_j}) \quad . \tag{3}$$

We use a Gaussian kernel:

$$K(\mathbf{d_i}, \mathbf{d_j}) = \exp\left(\frac{-(\mathbf{d}_i - \mathbf{d}j)^T (\mathbf{d}_i - \mathbf{d}_j)}{2\sigma^2} \right) - 1 \quad . \tag{4}$$

If the kernel function is a smooth function of the difference between data vectors, then it has a feature space made up of a series of sines and cosines, best explained in (Shawe-Taylor and Cristinini, 2005). The Fourier transform of the kernel function acts as a filter in feature space; if the σ of a Gaussian kernel is large, the kernel acts as a low-pass filter with small bandwidth. We use σ = 0.8. When data vectors have more than one dimension, the exponent

[1] If desired, the training data can be expanded by including synthetic data from multiple sources, for example, by using the density parameter to generate associated gravity data.

in Equation 4 decomposes; the feature space for multi-dimensional data is made of products of the sines and cosines of the elements of the data vectors.

A kernel can be used without any knowledge of its feature space, but understanding the feature space of the Gaussian kernel leads to the following insights:

- An SVM operating with a Gaussian kernel is interpolating between known models based on a low-pass filtered version of data space. The frequency dependence of Gaussian kernels can be adjusted to mirror that of band-limited seismic data.
- Non-linear relationships are captured by allowing frequencies in various dimensions of data space to interact with each other.
- Training data ought to adequately sample the data space in accordance with Fourier transform theorems.

The names for various SVM algorithms depend on the objective function used to find the α-coefficients. We use a least squares SVM (LS-SVM, Suykens et. al., 2002) which is also called kernel ridge regression. In all cases, a Gram matrix is constructed, $\Gamma_{i,j} = K(d_i, d_j)$. In a LS-SVM, this matrix is regularized and inverted. In a classic SVM it is used in a quadratic programming problem. The size of the problem depends on the number of training examples, not the size of the feature space or the number of desired model parameters.

1.2. Computation of the Gram matrix on a GPU

Computation of the Gram matrix is an inherently parallel problem and can be successfully implemented on a GPU. The training data is packed into 2-dimentional arrays of *colors*, where each color (red, green, blue, saturation) contains an element of a data vector. An output image is created on the GPU to store the Gram matrix. When a rectangle is drawn over this output image, the SVM kernel function executes over all of the pixels simultaneously. Figure 2 compares the performance of an INTEL Pentium M 2 Ghz, *vs* an nVidia 7800 Go chip. For gram matrices computed from 2000 training vectors with 2000 elements each, the GPU is 150 times faster than the CPU.

Fig. 2a) Fig. 2b)

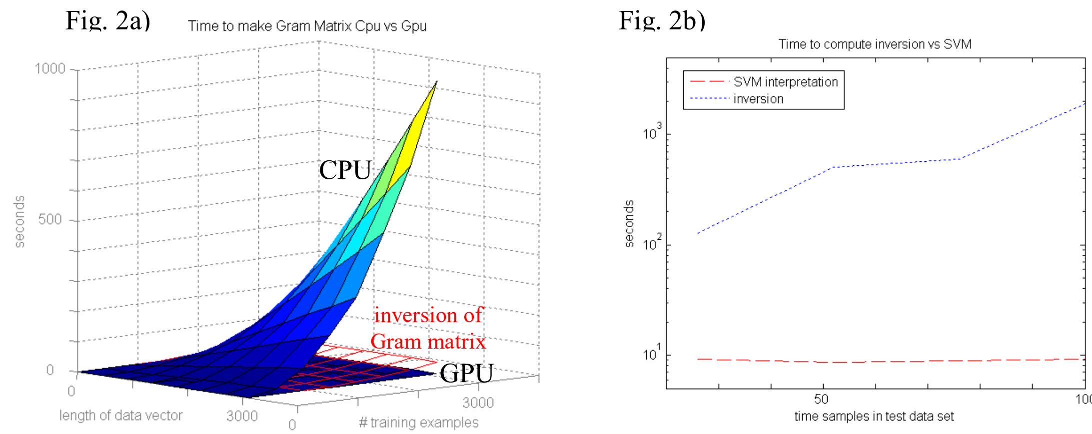

Figure 2: a) Computation of Gram matrix on CPU can take SVM 300 times longer than equivalent computation on GPU. b) The time it takes to invert a data set scales with the size of the data set, but the time it takes to interpret a data set using the SVM/GPU combination stays relatively constant.

2. Results

Results comparing the performance of a trained SVM relative to an inversion using a quasi-Newton method[2] are presented in Figure 1 and Table 1. The inversion was designed to minimize an objective function regularized using the same statistics that were sampled to make the training models for the SVM. The inversion, which minimizes data residuals does a better job of finding a model that can be used to fit the data, however the SVM does a better job of recovering an exact test model and is much faster than the inversion. The SVM is faster because it only requires about 1000 runs of the Zoeppritz equations in order to generate training data vs. the 30,000 runs required to compute the gradients of an iterative inversion, and because it can be easily implemented on the GPU. Figure 2b) illustrates that the time to perform an inversion scales with the size of a test data set, but the time to train and run an SVM stays the same.

3. Conclusions and future work

We have demonstrated the viability of the SVM inversion method and rationalized our choice of a Gaussian-based kernel in terms of its Fourier-like feature space. We are currently applying the method to more realistic synthetic data and to field data. We are implementing the complete SVM algorithm on a GPU and we are experimenting with using Genetic Algorithms combined with SVM inversion to optimize experimental design.

Acknowledgments: We would especially like to thank the IC Postdoctoral Fellowship program for supporting our project. This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

REFERENCES

Aki, K., and Richards P.G.,1980. *Quantitative seismology theory and methods*: W.H. Freeman and Co.

Castagne, J.P. and Backus, M.M., eds, 1993. *Offset-dependent reflectivity – theory and practice of AVO analysis*: SEG.

Dubrile, O., 2003. *Geostatistics for seismic data integration in earth models*: SEG.

Haykin, S., 1999. *Neural Networks 2nd. Ed.* : Prentice Hall.

Hidalgo, H., Leon, S.S., Gomez-Trevino, E., 2003. Application of the kernel method to the inverse geosounding problem. Neural Networks (16) : 349-353.

Kuzma, H. A., 2003. A Support Vector Machine for AVO Interpretation, 73rd Ann. Internat. Mtg.: SEG, 181-184.

Kuzma, H. A. and J. W. Rector, 2004. Non-linear AVO inversion using support vector machines: 74th Ann. Internat. Mtg.: SEG, 203-206.

Li, J. and Castagna J., 2003. Support Vector pattern recognition and AVO classification, 73rd Ann. Internat. Mtg.: SEG, 262-265.

Ohmer, J., Maire, F., Brown, R., 2005. Implementation of Kernel Methods on the GPU. Digital Image Computing: Techniques and Applications (DICTA'05), 78.

Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B. and Vandewalle, J., 2002. *Least Squares Support Vector Machines*: World Scientific Publishing.

Shawe-Taylor, J., and Cristianini, N., 2005. *Kernel Methods for Pattern Analysis* : Cambridge University Press, 65-66.

Vapnik, V., 1995. *The Nature of Statistical Learning Theory*, Springer-Verlag, 188.

[2] We used MATLAB 7.0.1.247074(R14) SP1, *fminunc* to perform the optimization.