HAL open science

Support vectors machines regression for estimation of Mars surface physical properties

Caroline Bernard-Michel, Sylvain Douté, Mathieu Fauvel, Laurent Gardes, Stéphane Girard

To cite this version:

Caroline Bernard-Michel, Sylvain Douté, Mathieu Fauvel, Laurent Gardes, Stéphane Girard. Support vectors machines regression for estimation of Mars surface physical properties. ESANN 2009 - 17th European Symposium on Artificial Neural Networks, Apr 2009, Bruges, Belgium. pp.195-200. hal-00761724v2

HAL Id: hal-00761724

https://hal.science/hal-00761724v2

Submitted on 6 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Support Vectors Machines Regression for Estimation of Mars Surface Physical Properties

C. Bernard-Michel[1], S. Douté[2], M. Fauvel[1], L. Gardes[1], and S. Girard[1] *

1- MISTIS, INRIA Rhône-Alpes & Laboratoire Jean Kuntzmann, Grenoble, France
2- Laboratoire de Planétologie de Grenoble, Grenoble, France

Abstract. In this paper, the estimation of physical properties from hyperspectral data with support vectors machines is addressed. Several kernel functions were used, from classical to advanced ones. The results are compared with Gaussian Regularized Sliced Inversion Regression and Partial Least Squares, both in terms of accuracy and complexity. Experiments on simulated data show that SVM produce highly accurate results, for some kernels, but with an increased processing time. Inversion of real images shows that SVM are robust and generalize well. In addition, the analysis of the support vectors allows to detect saturation of the physical model used to simulate data.

1 Introduction

Hyperspectral images are now a key tool for the analysis of remote planets [1]. The very high resolution in the spectral domain allows a fine characterization of the physical properties of the scene. For instance, the OMEGA sensor acquires the spectral radiance coming from the planet in more than 200 contiguous spectral channels. A pixel of the image is represented by a spectrum/vector $\mathbf{x} \in \mathbb{R}^d$, each component corresponds to a particular wavelength, d being the total number of wavelengths. Chemical composition, granularity, texture, and physical state are some of the parameters that characterize the morphology of spectra and thus the area of interest.

Deducing the physical parameters $\mathbf{y}$ from the observed spectra $\mathbf{x}$ is a central problem in classical physics, called an *inverse problem*. Since it generally cannot be solved analytically, optimization or statistical methods are necessary. Solving inverse problems requires an adequate understanding of the fundamental physics, *i.e.* a functional relation between $\mathbf{x}$ and $\mathbf{y}$ must be specified: $\mathbf{x} = g(\mathbf{y})$. Given g, different methods can be used to deduce the parameter from new observations. Current solutions to inverse problems can be divided into three main categories (for further details and comparisons, see [2]):

1. *Optimization algorithms*: These methods minimize an objective quality function that measures the fit between $\mathbf{x}$ and $g(\mathbf{y})$. Inverse problems are often ill-posed, therefore estimations can be unstable and a regularization is needed. For instance, a prior distribution on model parameters can be used.

*This work is supported by a contract with CNES through its Groupe Système Solaire Program and by INRIA and with the financial support of the "Agence Nationale de la Recherche" (French Research Agency) through its MDCO program ("Masse de Données et COnnaissances"). The Vahiné project was selected in 2007 under the reference ANR-07-MDCO-013.

2. *Look-up table (LUT) / k-nearest neighbors approach (k-NN)*: A large database (LUT) is generated by radiative transfer for many parameter values. Each observed spectrum is then compared with the LUT spectra in order to find the best match (the nearest neighbor), according to an objective function minimization. Parameters are then deduced from this best match.
3. *Training approaches*: A functional relation $\mathbf{y} = f(\mathbf{x})$ between spectra and parameters is assumed, such as $f^{-1} = g$, and a LUT is used to estimate f. For instance, training approaches include neural network approaches.

Inversion algorithms for hyperspectral images must be defined with the following constraints: *a*) Large datasets and various models require fast methodologies, *b*) High-dimensional data and "curse of dimensionality" with the associated sparsity issues require simple model with few parameters, *c*) Noisy spectra require robust algorithms. Optimization based approaches suffer from heavy computational cost and are thus not adapted to hyperspectral images. Nearest neighbors algorithms are faster, but the solution is unstable and noise-sensitive. Problems associated to training methods are the difficult interpretation of the estimated function f and the choice of the model. Moreover, neural networks are in general difficult to train for high dimensional data [3, Chapter 11].

The results presented in this paper follow previous research on hyperspectral image inversion [4]. Support Vectors Machines (SVM) are proposed to estimate the functional f. This training approaches is robust with respect to the dimensionality and has good generalization abilities, see [3, Chapter 12] or [5] for the classification of hyperspectral images. Using the *kernel trick*, the estimation of non-linear functions f is possible. Similarly to neural networks, the choice of the kernel function is not straightforward. In this article, several kernel functions were investigated, from linear to non-linear, leading to different levels of accuracy. For linear SVM, the connection with a dimension reduction technique, the Gaussian Regularized Sliced Inverse Regression (GRSIR) [4], is discussed. SVM are compared with GRSIR and Partial Least Square regression both in terms of accuracy and computing time.

SVM and the different kernels are presented in Section 2. Experimental results are reported in Section 3 and discussed in Section 4.

2 Support Vectors Machines Regression

2.1 Regression Problem

SVM are a supervised methods for regression or estimation stemming from the machine learning theory. For inversion problems, the algorithm, which is called the ϵ-SVR, approximates the functional $f : \mathbf{y} = f(\mathbf{x})$ using solutions of the form $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b$, where $\mathbf{x}_i$ are samples from the training set, k is a kernel function and $\big((\alpha_i)_{i=1,\dots,n},\ b\big)$ are the parameters of f which are found during the training process. The kernel k is used to produce non-linear functions. Given a training set, $(\mathbf{x}_i, \mathbf{y}_i)_{i=1,\dots,n} \in \mathbb{R}^d \times \mathbb{R}$, the training of an ϵ-SVR entails

the following optimization problem:

$$\min\left[\frac{1}{n}\sum_{i=1}^{n} l\big(f(\mathbf{x}_i),\mathbf{y}_i\big)+\lambda\|f\|^2\right] \text{ with } l\big(f(\mathbf{x}),\mathbf{y}\big)=\begin{cases} 0 \text{ if } |f(\mathbf{x})-\mathbf{y}|\leq\epsilon \\ |f(\mathbf{x})-\mathbf{y}|-\epsilon \text{ otherwise.}\end{cases}$$

The ϵ-SVR satisfies the sparsity constraint: Only some α_i are non-null which corresponding sample $\mathbf{x}_i$ are called "*Support Vectors*" (SVs). Some limitations come from the learning step: With a large training set, the processing time can be very long. Moreover, the problem is exacerbated when several optimizations for parameter selection are considered.

2.2 Kernel function

The choice of the kernel function is a crucial step with SVM. A kernel function is a similarity measure between two samples and corresponds to a dot product in some feature space. To be an acceptable kernel, the function should be positive semi-definite [3, Chapter 12]. In this work, several kernels were investigated:

- The linear kernel $k(\mathbf{x},\mathbf{z})=\langle\mathbf{x},\mathbf{z}\rangle$.
- The inhomogeneous polynomial kernel $k(\mathbf{x},\mathbf{z})=\big(\langle\mathbf{x},\mathbf{z}\rangle+q\big)^p$, where $q\geq 0$ induces a weight of each degree: $\big(\langle\mathbf{x},\mathbf{z}\rangle+q\big)^p=\sum_{k=0}^{p}\binom{p}{k}\langle\mathbf{x},\mathbf{z}\rangle^{p-k}q^k$.
- The Gaussian kernel $k(\mathbf{x},\mathbf{z})=\exp\big(-\gamma\|\mathbf{x}-\mathbf{z}\|^2\big)$.
- The spectral kernel $k(\mathbf{x},\mathbf{z})=\exp\big(-\gamma\alpha(\mathbf{x},\mathbf{z})^2\big)$, $\alpha(\mathbf{x},\mathbf{z})=\operatorname{acos}\big(\frac{\langle\mathbf{x},\mathbf{z}\rangle}{\|\mathbf{x}\|\|\mathbf{z}\|}\big)$.

The spectral kernel was first introduced in hyperspectral imagery for the classification purposes [6]. It is based on the angle between two spectra. It is a scale invariant kernel which is used in *k-nn* approaches or spectral unmixing [7].

Before running the algorithm, some hyperparameters need to be fitted: *a*) ϵ: Which controls the resolution of the estimation. Large values produce rough approximations while small values produce fine estimations. It can be set using some prior on the noise. *b*) λ: Which controls the smoothness of the solution. Large values imply nearly linear functions. *c*) Kernel parameters: γ for the Gaussian kernel, for instance. Cross-validation is used in this work to select the optimal parameters.

3 Experiments

3.1 Datasets

In this paper, real and simulated datasets are used. Real data have been collected during orbit 103 by the French imaging spectrometer OMEGA on board Mars Express Mission. A detailed analysis of this image by an expert led to a surface reflectance model. This model allows the generation of many synthetic spectra with the corresponding parameters: The proportions of CO_2, water and dust; and the grain sizes of water and CO_2. Centered multiGaussian noise has been added, its covariance matrix was determined experimentally from the real image. For the validation sake, separate training and testing datasets have been

Parameter	GRSIR	PLS	SVM						
			lin.	Gauss.	Spect.	0-Pol	0.5-Pol	1-Pol	2-Pol
Prop. of H_2O	0.28	0.32	0.31	0.14	0.25	0.24	0.17	0.14	**0.13**
Prop. of CO_2	0.19	0.31	0.30	**0.15**	0.27	0.27	0.18	0.16	**0.15**
Prop. of dust	0.11	0.22	0.22	**0.09**	0.19	0.19	0.11	0.10	0.10
Grain size of H_2O	0.34	0.39	0.39	**0.15**	0.34	0.33	0.23	0.19	0.18
Grain size of CO_2	0.16	0.24	0.25	**0.11**	0.21	0.20	0.14	0.12	**0.11**
CPU time (s)	**0.16**	0.66	3.57	10.30	5.89	5.98	10.20	60.30	478

Table 1: NRMSE and computing time for GRSIR, PLS and SVM with various kernels. "x-Pol" is $p = x$ in the polynomial kernel. The power of the polynomial kernel was fixed to 9 for each parameter, after cross-validation. The bottom line of the table corresponds to the training time of parameter "*Prop. of* H_2O" after the selection of optimal hyperparameters.

generated. The notations are the following: n (respectively n_t) is the number of spectra from the training data (respectively test data), $\mathbf{x}_i \in \mathbb{R}^d, i \in 1, \ldots, n$ denotes the spectra from the training data and $\mathbf{y}_i(p) \in \mathbb{R}, i \in 1, \ldots, n$ is one of the 5 associated parameters (respectively $\mathbf{x}_j^t,\ \mathbf{y}_j^t, j \in 1, \ldots, n_t$). In these experiments, $n = 3584$, $n_t = 3528$ and the number of spectral bands is $d = 184$. Each parameter $p = 1, \ldots, 5$ takes a finite number of values $\mathbf{y}(p)$ regularly distributed in a given interval of variation. The different realizations of the vector $\mathbf{y}_i$ are generated by building all possible combinations of the individual component values. In the following, the index p is omitted in $\mathbf{y}$ and in its associated functional f: $f_p(\mathbf{x}) = \mathbf{y}(p)$ is written $f(\mathbf{x}) = \mathbf{y}$.

3.2 Results

The quality of the estimation is assessed by computing the Normalized Root Mean Square Errors (NRMSE):

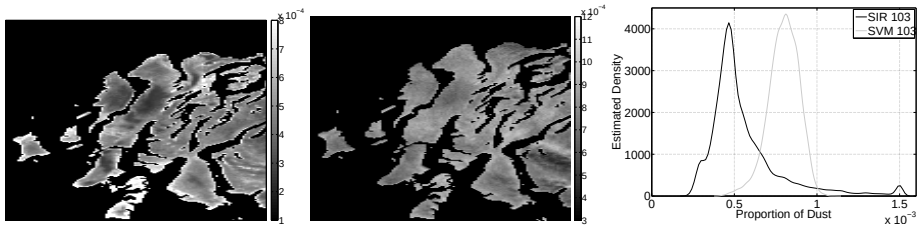
$$\text{NRMSE} = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}(\hat{\mathbf{y}}_i - \mathbf{y}_i)^2 \Big/ \frac{1}{n_t}\sum_{i=1}^{n_t}(\mathbf{y}_i - \overline{\mathbf{y}})^2} \text{ with } \overline{\mathbf{y}} = \frac{1}{n_t}\sum_{i=1}^{n_t}\mathbf{y}_i \tag{1}$$

where $\mathbf{y}_i$ is the real value and $\hat{\mathbf{y}}_i$ the estimated one. It is close to zero when the predicted values are accurate and becomes larger when predictions are poor. SVM regression is compared with GRSIR and PLS. These methods are "training approaches", GRSIR and PLS both first reduce the dimension of the data using $\mathbf{x}$ and $\mathbf{y}$, and then perform the estimation in the reduced space. PLS is a linear estimator while GRSIR is a piecewise linear one, for details see [8, 9].

From Table 1, it is clear that the worst results[1] in terms of accuracy are obtained with linear algorithms: PLS and SVM with the linear kernel. Best accuracy are obtained with the SVM-Gaussian kernel, followed by SVM-inhomogeneous polynomial kernels with $q \geq 1$. However, the last ones require more time to be trained. SVM with spectral or homogeneous polynomial kernel yield the lowest accuracy. Considering the training time, SVM lead to longer processing times, for every kernels.

Inversions are performed on the real image using the functional f. Results are given in Fig. 1, for GRSIR and SVM-Gaussian kernel. The histogram of the

[1]Results for GRSIR and PLS differ from [4] because the selection of optimal parameters is done in a different way, cross-validation in this article, and the simulated datasets are different.



(a) (b) (c)

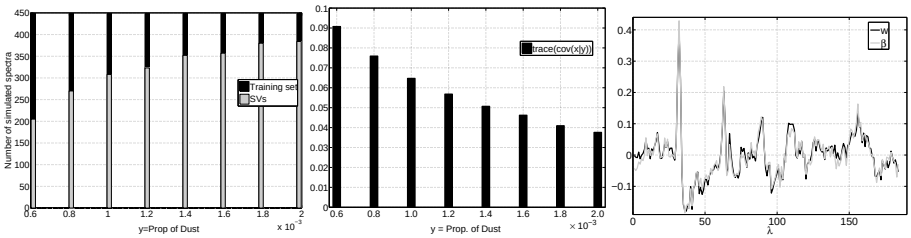
Fig. 1: Proportion of dust estimated by (a) GRSIR, (b) SVM with a Gaussian kernel from the hyperspectral image observed from orbit 103 and (c) is the histogram of (a) and (b).

estimated values is also reported and shows a shift between GRSIR and SVM estimates (astrophysical interpretations are given in [10]). However, since no ground data is available, the inversion accuracy of real image is difficult to appreciate.

4 Discussion

With SVM, the training quality can be assessed by looking at the Support Vectors (SVs). Selection of many SVs may indicate that the optimization problem is hard (but not necessarily badly solved). Fig. 2.(a) shows the proportion of $\mathbf{y}$ corresponding to SVs (light gray) from the training set (dark gray). The fraction increases as $\mathbf{y}$ increases too. To understand why the problem becomes harder, the spectral variance of the simulated spectra as a function of the wavelength is reported in Fig. 2.(b). It is computed as the trace of the covariance matrix of samples $\mathbf{x}$ for which $\mathbf{y}$ is constant. It can be seen that a high $\mathbf{y}$ value leads to: on the one hand a small variance of the simulated spectra, see Fig. 2.(b), and, on the other, a high number of SVs, see Fig. 2.(a). This phenomenon is explained by the properties of the physical model: For some values, the model saturates. Consequently when the saturation is reached, increasing the values of the parameters has little consequences on the simulated spectra (and the variance remains low) thus making the estimation more difficult, due to strong non-linearity between $\mathbf{x}$ and $\mathbf{y}$. The main practical implication is in the generation of the training samples: Looking at the SVs after a first training provides information to possibly generate new samples for which the estimation is difficult. Finally, note that SVM and other training methods face difficulties in estimating f when saturation is reached.

Unlike statistical methods such as GRSIR, interpretation of SVM's functional estimation is not possible, except for the linear kernel: $f(\mathbf{x}) = \sum_i^n \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b = \langle \mathbf{x}, \sum_i^n \alpha_i \mathbf{x}_i \rangle + b = \langle \mathbf{x}, \mathbf{w} \rangle + b$. Surprisingly, the vector $\mathbf{w}$ is akin to the vector β estimated with GRSIR [4], see Fig.2.(c). In GRSIR, β is used to reduce the dimension and the estimation f' is done in the reduced space: $f(\mathbf{x}) = f'(\langle \mathbf{x}, \beta \rangle)$. It explains why GRSIR performs better than linear SVM: The estimation is done in the one-dimensional space spanned by β (respectively $\mathbf{w}$) with a piecewise linear estimator (respectively linear estimator). The precise theoretical comparison is omitted here for the sake of conciseness.



(a) (b) (c)

Fig. 2: (a) Proportion of $\mathbf{y}$ (light gray) corresponding to SVs (Gaussian kernel) from the training set (dark gray), $\mathbf{y}$ was "*Prop. of dust*"; (b) Spectral variance of simulated $\mathbf{x}$ as function of $\mathbf{y}$, $\mathbf{y}(p)$ was "*Prop. of dust*" ; (c) GRSIR axis β and SVM (linear kernel) normal vector $\mathbf{w}$ as function of the wavelength λ.

As a conclusion of this work, SVM with non-linear kernels produce accurate results for the inversion of hyperspectral images. Experiments on real data images confirm their robustness to the training set and to the high dimensionality of the data. However, the training time can be dramatically increased for some kernel, making the use of SVM in a practical situations sensitive. Current research is now oriented in the definition of kernel handling more efficiently the saturation in the physical model.

References

[1] R. W. Carlson, P. R. Weissman, W. D. Smythe, J. C. Mahoney, the NIMS Science, and Engineering Teams. Near infrared spectrometer experiment on Galileo. *Space Science Reviews*, 60:457–502, 1992.

[2] D.S. Kimes, Y. Knyazikhin, J.L. Privette, A.A. Abuegasim, and F. Gao. Inversion methods for physically-based models. *Remote Sensing Reviews*, 18:381–439, 2000.

[3] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2003.

[4] C. Bernard-Michel, S. Douté, L. Gardes, and S. Girard. Inverting hyperspectral images with gaussian regularized sliced inverse regression. In *European Symposium on Artificial Neural Networks, Advances in Computational Intelligence and Learning*, 2008.

[5] A. Plaza, J. A. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, J. C. Tilton, and G. Triani. Recent advances in techniques for hyperspectral image processing. *Remote Sensing Environment*, To appear, 2009.

[6] M. Fauvel, J. Chanussot, and J. A. Benediktsson. Evaluation of kernels for multiclass classification of hyperspectral remote sensing data. In *IEEE ICASSP'06*, May 2006.

[7] N. Keshava. Distance metrics and band selection in hyperspectral processing with application to material identification and spectral librairies. *IEEE Trans. Geosci. Remote Sens.*, 42:1552–1565, July 2004.

[8] C. Bernard-Michel, L. Gardes, and S. Girard. Gaussian regularized sliced inverse regression. *Statistics and Computing*, 19:85–98, 2009.

[9] C. Bernard-Michel, L. Gardes, and S. Girard. A note on sliced inverse regression with regularizations. *Biometrics*, 64:982–986, 2008.

[10] C. Bernard-Michel, S. Douté, M. Fauvel, L. Gardes, and S. Girard. Retrieval of mars surface physical properties from omega hyperspectral images using regularized sliced inverse regression. *Journal of Geophysical Research - Planets*, 114, 2009. E06005.