Machine learning techniques for the inversion of planetary hyperspectral images

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

MACHINE LEARNING TECHNIQUES FOR THE INVERSION OF PLANETARY HYPERSPECTRAL IMAGES

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 physical analysis of planetary hyperspectral images is addressed. To deal with high dimensional spaces (image cubes present 256 bands), two methods are proposed. The first method is the support vectors machines regression (SVM-R) which applies the structural risk minimization to perform a non-linear regression. Several kernels are investigated in this work. The second method is the Gaussian regularized sliced inverse regression (GRSIR). It is a two step strategy; the data are map onto a lower dimensional vector space where the regression is performed. Experimental results on simulated data sets have showed that the SVM-R is the most accurate method. However, when dealing with real data sets, the GRSIR gives the most interpretable results.

***Index Terms*—** Hyperspectral images, Gaussian regularized sliced inversion regression, SVM, Mars surface.

1. INTRODUCTION

For two decades, imaging spectroscopy has been a key technique for exploring planets [1, 2]. 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 y from the observed spectra $\mathbf{x}$ is a central problem in geophysics, 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 physics of the signal formation, *i.e.* a functional relation between $\mathbf{x}$ and y must be specified: $\mathbf{x} = g(y)$. Given g, different methods can be used to deduce the parameters from new observations. Current solutions to inverse problems can be divided into three main categories (for further details and comparisons, see [3]):

1. *Optimization algorithms*: These methods minimize an objective quality function that measures the fit between $\mathbf{x}$ and $g(y)$. 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.
2. *Look-up table (LUT) / k-nearest neighbors approach (k-NN)*: A large database (LUT) is generated by a physical model 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 $y = f(\mathbf{x})$ 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 statistical model. Moreover, neural networks are in general difficult to train for high dimensional data [4, Chapter 11].

In this paper, several approaches are presented to estimate the functional f: The well-known Support Vectors Machines regression (SVM-R) [4, Chapter 12], which works in high dimension, the Gaussian Regularized Sliced Inverse Regression (GRSIR) [5], which reduces the dimension before the estimation, the Partial Least Squares regression [4, Chapter 3] and the k-NN [4, Chapter 13]. The starting point was that methods not relying on statistical models (SVM or k-NN) have two main advantages over parametric ones: no prior information is needed and no parameters estimation are necessary. However, in general, results are hardly interpretable and thus no information about the relationship between the input and the output is available. On the contrary, methods such as GRSIR or PLS reduce the dimension of spectra and the resulting subspace provides some physical information which can be used by astrophysicists [6]. Moreover, the training time is generally favorable to the parametric methods.

Section 2 presents the different methods. Emphasis is put on

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.

SVM and GRSIR which are novel methods in planetary images analysis. The data sets are presented in Section 3 and the experimental results are given in Section 4. The analysis of real observation is done in Section 5.

2. INVERSION METHODS

For each method, we are given a training set $\big(\mathbf{x}_i, y_i\big)_{i=1}^n \in \mathbb{R}^d \times \mathbb{R}$ and we try to learn $f : y = f(\mathbf{x})$. We concentrate on $\mathbb{R}$-valued function, *i.e.*, p functions are necessary to deduce p parameters from the spectra.

2.1. Support Vectors Machines Regression

SVM are 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 using solutions of the form $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b$, where 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, the training of an ϵ-SVR entails the following optimization problem:

$$\min\left[\frac{1}{n}\sum_{i=1}^n l\big(f(\mathbf{x}_i), y_i\big) + \lambda\|f\|^2\right] \tag{1}$$

$$\text{with } l\big(f(\mathbf{x}), y\big) = \begin{cases} 0 \text{ if } |f(\mathbf{x}) - y| \leq \epsilon \\ |f(\mathbf{x}) - y| - \epsilon \text{ otherwise.} \end{cases}$$

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 training time can be very long. Moreover, the problem is exacerbated when several optimizations for parameter selection are considered.

The choice of the kernel function is a crucial step with ϵ-SVR. 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 [4, Chapter 12]. In this work, several kernels were investigated:

- Linear kernel $k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z}\rangle$.
- 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: $k(\mathbf{x}, \mathbf{z}) = \sum_{k=0}^p \binom{p}{k} \langle \mathbf{x}, \mathbf{z}\rangle^{p-k} q^k$.
- Gaussian kernel $k(\mathbf{x}, \mathbf{z}) = \exp\big(-\gamma\|\mathbf{x} - \mathbf{z}\|^2\big)$.
- Spectral kernel $k(\mathbf{x}, \mathbf{z}) = \exp\big(-\gamma\alpha(\mathbf{x}, \mathbf{z})^2\big)$, with $\alpha(\mathbf{x}, \mathbf{z}) = \operatorname{acos}\big(\frac{\langle \mathbf{x}, \mathbf{z}\rangle}{\|\mathbf{x}\|\|\mathbf{z}\|}\big)$.

The spectral kernel was first introduced in hyperspectral imagery for the classification purposes [7]. 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 [8].

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: *e.g.*, γ for the Gaussian kernel.

Table 1. SIR Regularization

φ	m	Ω	Eigen problem	Regularization
$\frac{1}{\delta_i}$	$= d$	Σ^{-1}	$\Sigma^{-1}\Gamma$	-
1	$= d$	$\mathbf{I}_d$	$\big(\Sigma + \tau\mathbf{I}_d\big)^{-1}\Gamma$	Ridge
1	$< d$	$\sum_{i=1}^m \mathbf{v}_i\mathbf{v}_i^t$	eq. (3)	PCA-Ridge
δ_i	$= d$	Σ	$\big(\Sigma^2 + \tau\mathbf{I}_d\big)^{-1}\Sigma\Gamma$	Tikhonov
δ_i	$< d$	$\sum_{i=1}^m \delta_i\mathbf{v}_i\mathbf{v}_i^t$	eq. (3)	PCA-Tikhonov

2.2. Gaussian Regularized Sliced Inverse Regression

To circumvent the "curse of dimensionality" effects, an alternative approach is to reduce the dimension of the data before the estimation. This is done by mapping the data onto a lower dimensional space and then doing the estimation:

$$y = f\big(\boldsymbol{\beta}^t\mathbf{x}\big). \tag{2}$$

where $\boldsymbol{\beta}^t\mathbf{x}$ denotes the projection on the subspace spanned by $\boldsymbol{\beta}$.

The Principal Component Analysis (PCA) is surely one of the most used approach: $\boldsymbol{\beta}$ corresponds to the p firsts eigenvectors of the covariance matrix Σ of $\mathbf{x}$. However, in the case of a regression problem, PCA is generally not satisfying since only the explanatory variables $\mathbf{x}$ are considered and the dependent variable y is not taken into account. Specific dimension reduction techniques have been developed for regression problems, and among them Sliced Inverse Regression (SIR) is very effective in high dimensional spaces [9]. It consists of applying PCA to the inverse regression curve $\mathbb{E}(\mathbf{x}|y)$ instead of applying it to the original predictor $\mathbf{x}$. The projection axis $\boldsymbol{\beta}$ is then deduced by calculating the eigenvector corresponding to the largest eigenvalue of $\Sigma^{-1}\Gamma$, where $\Gamma = \operatorname{Cov}(\mathbb{E}(\mathbf{x}|y))$ is the covariance matrix of the inverse regression curve.

In high dimensional vector spaces, inverse problems are generally ill-posed [10, 11], *i.e.*, Σ is ill-conditioned making its inversion difficult. We thus have proposed to compute a Gaussian Regularized version of Sliced Inverse Regression (GRSIR). Theoretical details can be found in [5]. The concept of this method is to incorporate some Gaussian prior on the projections in order to dampen the effect of noise in the input data, and to make the solution more regular or smooth. The ill-posed problem is then replaced by a slightly perturbed well-posed problem. Finally, GRSIR consists of computing the eigenvector corresponding to the largest eigenvalue of

$$\big(\Omega\Sigma + \tau\mathbf{I}_d\big)^{-1}\Omega\Gamma \tag{3}$$

where τ is a positive regularization parameter, $\mathbf{I}_d$ is the identity matrix and Ω is a $d \times d$ matrix modeling the prior on the projection: it describes which directions are the most likely to contain $\boldsymbol{\beta}$ [5].

Using eigenvalue decomposition of Σ, we have proposed several definitions of Ω that lead to several well known regularizations. Let us write

$$\Sigma = \sum_{i=1}^d \delta_i\mathbf{v}_i\mathbf{v}_i^t \tag{4}$$

with $\delta_1 \geq \ldots \geq \delta_d$, the eigenvalues and $\mathbf{v}_i$ their associated eigenvectors. Then for all real valued function φ, Ω is defined as:

$$\Omega(\varphi) = \sum_{i=1}^m \varphi(\delta_i)\mathbf{v}_i\mathbf{v}_i^t \tag{5}$$

Table 2. NRMSE and computing time for GRSIR, PLS and SVM with various kernels. "x-Pol" is $q = x$ 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.

Parameter	GRSIR				PLS	SVM						
	Ridge	Tik.	PCA-Ridge	PCA-Tik.		lin.	Gauss.	Spect.	0-Pol	0.5-Pol	1-Pol	2-Pol
Prop. of H_2	0.29	0.29	0.30	0.29	0.32	0.31	0.14	0.25	0.24	0.17	0.14	**0.13**
Prop. of CO_2	0.20	0.20	0.22	0.22	0.31	0.30	**0.15**	0.27	0.27	0.18	0.16	**0.15**
Prop. of dust	0.11	0.12	0.12	0.12	0.22	0.22	**0.09**	0.19	0.19	0.11	0.10	0.10
Grain size of H_2O	0.34	0.34	0.34	0.35	0.39	0.39	**0.15**	0.34	0.33	0.23	0.19	0.18
Grain size of CO_2	0.17	0.17	0.17	0.17	0.24	0.25	**0.11**	0.21	0.20	0.14	0.12	**0.11**
CPU time (s)	0.17	0.18	0.20	0.20	0.66	3.57	10.30	5.89	5.98	10.20	60.30	478

with $m \in \{1, \ldots, d\}$. Table 1 sums up the different strategies we have proposed. φ controls which directions of Σ that are favored: For instance, with the classical SIR approach (first row of Table 1) direction corresponding to small variances are most likely, while no directions are privileged with ridge regularization. For Tikhonov regularization, directions corresponding to large variance are most likely, in contrast to classical SIR. PCA based regularization approaches correspond to the situation where only directions with large variance are kept, *i.e.*, a dimension reduction of Σ is done. Once $\boldsymbol{\beta}$ is computed, a piecewise linear estimator is used, *i.e.* f in eq. (2) is a 1-dimensional piecewise linear function.

3. DATA SETS

In this paper, real and simulated data sets are used. Real data have been collected during orbit 41 61 and 103 by the imaging spectrometer OMEGA on board Mars Express Mission. A detailed analysis of this image by an expert led to a surface reflectance model [1]. This model allows by radiative transfer calculations the generation of many synthetic spectra with the corresponding physical 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 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 $y_i^p \in \mathbb{R}, i \in 1, \ldots, n, p \in 1, \ldots, 5$, is one of the 5 associated parameters (respectively $\breve{\mathbf{x}}_j$, $\breve{y}_j^p, j \in 1, \ldots, n_t$). In these experiments, $n = 3584$, $n_t = 3528$ and the number of spectral bands is $d = 184$. Each parameter takes a finite number of values regularly distributed in a given interval of variation. The different realizations of the vector y_i^p are generated by building all possible combinations of the individual parameter values. In the following, the index p is omitted in y and in its associated functional f: $f_p(\mathbf{x}) = y^p$ is written $f(\mathbf{x}) = y$.

For validation purpose, see Section 4, spectra from several images of the same area of Mars where extracted. They correspond to three acquisitions at different times. All the spectra were atmospherically corrected.

4. EXPERIMENTS

In all experiments, parameters were selected by a 5-fold cross validation. The quality of the estimation is assessed by computing the Normalized Root Mean Square Errors (NRMSE):

$$\text{NRMSE} = \sqrt{\frac{\frac{1}{n_t}\sum_{i=1}^{n_t}(\hat{y}_i - \breve{y}_i)^2}{\frac{1}{n_t}\sum_{i=1}^{n_t}(\breve{y}_i - \overline{y})^2}} \text{ with } \overline{y} = \frac{1}{n_t}\sum_{i=1}^{n_t}\breve{y}_i \qquad (6)$$

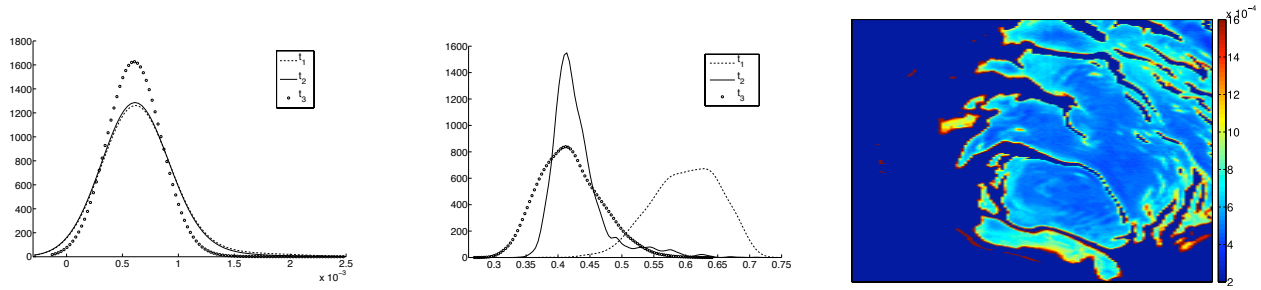
where $\breve{y}_i$ is the real value and $\hat{y}_i$ the estimated one. NRMSE is close to zero when the predicted values are accurate and becomes larger when predictions are poor. Results are reported in Table 2.

From the table, the worst results in terms of NRMSE are obtained with linear algorithms: PLS and SVM with the linear kernel. Non-linear SVM provides the best results in terms of accuracy, for both Gaussian and inhomogeneous polynomial kernels. Spectral or homogeneous polynomial kernel provide the lowest accuracy for non-linear SVM. For the GRSIR approaches, results are nearly the same, and less accurate than non-linear SVM. The different regularization methods yield the same results, however regularization parameters (τ and m) are set to different values: Using PCA based regularization, τ is in general set to a lower value than with non-PCA based regularization. Furthermore, $\boldsymbol{\beta}$ is not the same depending on the regularization. For instance, $\cos\left(\boldsymbol{\beta}_{\text{ridge}}, \boldsymbol{\beta}_{\text{pca-ridge}}\right) = 0.87$ and $\cos\left(\boldsymbol{\beta}_{\text{ridge}}, \boldsymbol{\beta}_{\text{Tik.}}\right) = 0.93$.

Considering the training time, SVM based approaches lead to the longer processing times, for every kernel and GRSIR is the fastest. Because of the diagonalization of the covariance matrix in eq. (5), PCA based regularization is a little longer. Note that the SVM solver was the LIBSVM, which is a higly optimized C++ solver, while the GRSIR was naively implemented in Matlab. Thus the difference may be higher after a proper implementation of GRSIR.

Inversions are performed on the real data using the functional f. We have considered only the best SVM (Gaussian kernel) and the best GRSIR (ridge). Since no ground data is available, the inversion accuracy is difficult to appreciate quantitatively. We used two subjective criteria: the physical possibility, *e.g.*, proportion is greater than 0 and lower than 1, and the stability of estimations over the time, *e.g.*, estimations from the same area at closed dates should be closed.

Fig. 1.(a) and Fig. 1.(b) present the estimation of the "*proportion of dust*". It can be seen that estimations are very different from GRSIR and SVM. Estimates provided by GRSIR are more realistic: First, the range is around 10^{-3} which is physically acceptable; second, the variation of the proportion between two dates is low in agreement with experts analysis. On the opposite, SVM estimations are not convincing since the range of the estimations is not realistic. This phenomenon can be mitigated by a suitable selection of the simulated training set, as detailed in [12, 13]. The idea is to select training samples that are similar to those to be inverted. However,



(a) GRSIR (b) SVM (c) Orbit 41

Fig. 1. (a) GRSIR and (b) SVM: Histogram of the "*proportion of dust*" estimates from real data from the same geographical area acquired at different dates (t_1, t_2 and t_3). (c) Estimation of the "*proportion of dust*" of orbit 41.

this step is time consuming and still does not provide satisfactory results with SVM.

Fig. 1.(c) presents the observation 41 by GRISR for the parameter "*proportion of dust*". The mapping is very smooth and the proportion of dust increases significantly with proximity to the boundaries.

5. DISCUSSION

A machine learning discussion is done in this section. We let readers interested in a detailed astrophysical analysis consult the reference [6].

Non-linear SVMs provide very accurate results in term of NRMSE on the simulated data sets. It comes with an increased training time, which can be critical with the polynomial kernel. Therefore, Gaussian kernel should be preferably used. The definition of new kernels handling more efficiently the physical model is under investigation. The GRSIR approach provides less accurate results on the simulated data sets, but it performs better than SVM of real data sets and his training is very fast.

The regularization strategy does not influence too much the results once the optimal parameters have been found, the key point is to regularize. In the experiments, all the parameters have been selected by cross-validation using the simulated data sets. In the context of our work (training on simulated data and validation on real data), this strategy should be changed. Statistical differences exist between the simulated data and real data, depending on the geometry of the observed image and the atmospheric effects. For instance, the cosine angle, computed with the Frobenius inner product, between the covariance matrix Σ of simulated data and the real data is 0.60. We are currently working on a semi-supervised framework to match statistics from simulated data and real data before the estimation of the regression function. This should improve the results obtained on real data sets.

As a conclusion, handling efficiently the inversion of hyperspectral images is possible and of an interest for astrophysicists. First results are promising yet some works are still needed to obtain a fully automatised framework.

6. REFERENCES

[1] S. Douté, B. Schmitt, R. M. C. Lopes-Gautier, R. W. Carlson, L. Soderblom, and J. Shirley, "Mapping SO_2 frost on Io by the modeling of NIMS hyperspectral images," *Icarus*, vol. 149, pp. 107–132, 2001.

[2] 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*, vol. 60, pp. 457–502, 1992.

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

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

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

[6] 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*, To appear, 2009.

[7] 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.

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

[9] K.C. Li, "Sliced inverse regression for dimension reduction," *Journal of the American Statistical Association*, vol. 86, pp. 316–327, 1991.

[10] A. Tarantola, *Inverse problem theory and model parameter estimation*, SIAM, 2005.

[11] R.C. Aster, B. Borchers, and C.H. Thurber, *Parameter Estimation and Inverse Problems*, Elsevier Academic Press, 2005.

[12] 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.

[13] C. Bernard-Michel, S. Douté, M. Fauvel, L. Gardes, and S. Girard, "Support vectors machines regression for the estimation of Mars surface physical properties," in *European Symposium on Artificial Neural Networks, Advances in Computational Intelligence and Learning*, 2009.