



Genetic programming to extract features from the whole-sky camera for cloud type classification

J. Huertas¹, J. Rodríguez-Benítez², D. Pozo², R. Aler¹, Inés M. Galván¹

¹Computer Science Department EVANNAI, University of Carlos III Avda. Unividersidad, 30, 28911 *Leganés* (Spain) Phone number:+0034 916 248879, e-mail: <u>jahuerta@inf.uc3m.es</u>

> ²Physics Department MATRAS, University of Jaén

Abstract In the automatic cloud classification problem it is very important to extract relevant features from the cloud images that can be used as inputs to the classifiers. Typically, sets of hand-designed features, based on the red, green, and blue channels, are used. For instance, spectral and textural, among other characteristics, are commonly extracted from cloud images. Genetic Programming is a powerful tool that has been used to automatically generate functions in a variety of problems. In this work, it is proposed to use Genetic Programming to automatically construct image features for cloud classification. Specifically, the constructed function aims to transform an image, pixel by pixel, and then computing the mean and the standard deviation of the transformed image. The performance of this method is measured against a set of expertdefined features. Experiments have been carried out on a database of whole-sky cloud images. Results show that the proposed method is able to achieve a similar accuracy as the 4 most important features from the expert feature-set.

Keywords

Genetic Programming, Feature Extraction, Cloud classification, Whole-sky images

1. Introduction

When working with solar irradiation, there are several sky conditions that may interact with sunrays, affecting the production of electrical energy in solar power plants. Among the set of possible interferences, clouds produce the most variability in the solar energy output of a given solar panel. Clouds are a persistent problem in solar forecasting [1], due to extreme variety in shape, thickness, size and sky coverage. Another factor in their problematic effects is the visual similarity between clouds, and although they can be classified by experts, it is costly and time-consuming to do so in large datasets of cloud images [2] In order to predict the type of a cloud many tools can be used, radiometer, thermometer, ceilometer, etc. However the most basic and intuitive is a ground based, whole-sky camera, which is economic and flexible for cloud-related problems [3]. Statistics and parameters can be computed on the camera channels providing characteristics such as cloud shape, texture or the percentage of sky covered by the cloud. The type and number of statistics has increased in the last years [4]-[6] with the aim to provide distinctive statistics to classify the cloud type.

The set of statistics or features can be used as inputs for machine learning algorithms to automatic cloud type classification. Different techniques are used to solve this problem, one of them is the k-Nearest Neighbors in [5] and [6] which is common practice in the cloud classification task. Other authors have taken different approaches, for example using Neural Networks and Support Vector Machines as in [7]. Classifying clouds has proven to be a difficult problem, without a reliable high-accuracy solution yet.

This article presents the problem of automatically constructing new statistics or features for cloud classification using Genetic Programming (GP) [8]. GP is a kind of genetic algorithm that can be used to automatically evolve functions, which has also been applied to the feature discovery problem, such as views from vehicles [9], known machine learning benchmarks [10] or vibration data for fault classification [11]. GP has proven useful to design features automatically for their domains better than their expert-designed counterparts. Here, it is proposed to apply GP to discover functions that transform cloud images, represented as red, green, and blue matrices, into features that can be used by machine learning classification algorithms. The GP evolved features will be compared with the set of statistics proposed by Heinle [6], which are widely used in the cloud classification problem.

This article opens with a brief introduction to GP (section 2), followed by the description of the Heinle feature set used in the final comparison (section 3). Section 4 describes the GP feature extraction method proposed in this work. The experimentation performed over the generated features is detailed, and evaluated in section 5. The paper closes with the conclusions and future work section.

2. Genetic Programming

Evolutionary algorithms is a family of optimization methods based on biological evolution [12]. These algorithms optimize a population of candidate solutions (or individuals) for a given optimization function (called 'fitness'), using genetic operators (mainly, crossover and mutation). The crossover operation randomly takes parts of a pair of individuals (or candidate solutions) to form a third new individual in a population. The mutation operation randomly changes parts of a single individual. After performing the crossing and mutation operations are made, a new generation is created.

GP is based on genetic algorithms. Individuals are defined as trees, where leaf nodes are variables or constants and every other node is an operator taken from an operator set. The operator set defines which operations are available for constructing individuals. A flowchart of GP can be found on (Fig. 1)



Fig. 1. GP typical procedure representation

3. Feature extraction for cloud classification

Automatic cloud classification can be carried out by means of machine learning algorithms using as inputs to the models a set of features extracted from the images. The aim of this section is to describe a set of features that have been applied successfully for cloud classification [6] and that will be used in this article as benchmark to compare with our own approach.

Table I. - Heinle's features for cloud classification.

FEATURE	TYPE	FORMULA
μ' ; Red	Spectral	$1 \sum_{n=1}^{n} \sum_{n=1}^{n} 1$
average		$\mu^c = \frac{1}{n^2} \sum_{j=0}^{\infty} \sum_{i=0}^{\infty} M^c_{i,j}$
μ^{b} ; Blue average	Spectral	Same as above
σ^{b} ; Blue deviation	Spectral	$\sigma^{c} = \sqrt{\frac{1}{n^{2}} \sum_{j=0}^{n} \sum_{i=0}^{n} (M_{i,j}^{c} - \mu^{c})^{2}}$
γ^b ; Blue skewness	Spectral	$\gamma^{c} = \frac{1}{n^{2}} \sum_{j=0}^{n} \sum_{i=0}^{n} \left(\frac{M_{i,j}^{c} - \mu^{c}}{\sigma^{c}} \right)^{3}$
<i>D</i> ^{rg} ; Red – Green mean difference	Spectral	$D^{c_1 c_2} = \mu^{c_1} - \mu^{c_2}$
<i>D</i> ^{<i>rb</i>} ; Red – Blue difference	Spectral	Same as above
<i>D^{gb};</i> Green – Blue mean Difference	Spectral	Same as above
<i>EN^b</i> ; Blue	Textural	$EN^{c} = \sum_{j=0}^{g} \sum_{i=0}^{g} [p_{i,j}^{c}]^{2}$
ENT ^b ; Blue	Textural	$ENT^{c} = \sum_{j=0}^{g} \sum_{i=0}^{g} p_{i,j}^{c} \log_{2} p_{i,j}^{c}$
<i>CON</i> [®] ; Blue	Textural	$CON^{c} = \sum_{j=0}^{g} \sum_{i=0}^{g} (i-j)^{2} p_{i,j}^{c}$
<i>HOM^b;</i> Blue	Textural	$HOM^{c} = \sum_{j=0}^{g} \sum_{i=0}^{g} \frac{p_{i,j}^{c}}{1 + a-b }$
C; % cloud coverage	Coverage	$\frac{M_{i,j}^r}{M_{i,j}^b} > T$; $C = \frac{cp}{tp}$; $T = 0.82$

Heinle's features are obtained from the red, green, and blue channels of an image. These channels are represented using three matrices M^r , M^g , M^b , red, green and blue respectively, with integer values between 0 and 255. Each (i, j) location in the matrices corresponds to a pixel in the image. There are several types of image features (Table I): spectral features, textural features, and cloud coverage. Spectral features use the color matrix M^c exclusively (where *c* can be red, green or blue), extracting statistical measures directly from it. These are the simplest from the feature set and require very little processing to obtain. The textural features make use of a Grey Level Co-occurrence Matrix (GLCM). This is a transformation over one of the color channels. The result is a

square matrix with size $g \ x \ g, \ g$ being the number of gray levels considered in the image. Every element of the GLCMs $(p_{i,j}^c)$ represents the relative frequency that two adjacent pixels have the same value in a set direction, and *c* represents the color of the source channel. Here g = 256 is used, and the direction is oriented to the right. Finally, cloud coverage is the percentage of sky covered by a cloud. It is computed by means of a threshold applied on the red and blue channels.

4. Genetic Programming for feature extraction

In this section, the GP-based approach to extract features for cloud classification is described. It is proposed to use GP for finding a function f(r,g,b) that transforms one pixel, (represented by its red, green, and blue components) into a single real value. f is applied to each of the pixels in the image (represented by matrices M^r , M^g , M^b), obtaining a transformed image represented by matrix M. From M two values are computed: the average (μ^{gen}) and the standard deviation (σ^{gen}), given by equations (1) and (2). These are the estimated image features that will be used for the cloud classification.

$$\mu^{gen} = \frac{1}{n^2} \sum_{j=0}^{n} \sum_{i=0}^{n} M_{i,j}^c \tag{1}$$

$$\sigma^{gen} = \sqrt{\frac{1}{n^2} \sum_{j=0}^{n} \sum_{i=0}^{n} (M_{i,j}^c - \mu^{gen})^2} \quad (2)$$

where $M_{ij} = f(M_{i,j}^r, M_{i,j}^g, M_{i,j}^b)$

Three elements need to be defined to use GP: the terminals, the operator set, and the fitness function. The terminals are either the pixel values (r, g, and b) or the constants that can appear in function f (which in GP are values randomly generated when the initial population is created). The operator set contains the basic functions that GP can use to construct function f. These include several arithmetic operations (+, -, *, /, log, min, max) and two logical operations (< and >). The later, return 0 or 1, when they are false or true, respectively.

The fitness function is meant to estimate the accuracy of function f. Fitness computation is done in two steps: first, μ^{gen} and σ^{gen} are computed (see Eq. 1), and second, a linear SVM (LSVM) is trained and tested on the cloud dataset, but using only μ^{gen} and σ^{gen} as input features. The dataset is divided into a training set and a validation set with a 90% partition ratio, balanced classes and trying to group temporally close instances in the same set (either training or validation). The validation accuracy of the LSVM is the fitness of function f. In this work GP generates an initial population of 2000 individuals that iterates over 100 generations where the 200 best individuals are considered the elite. At the end of the process the individual in the elite with the highest fitness is selected. Sometimes two different individuals might be identical in fitness value (validation accuracy). When this happens, the validation macro-average accuracy (average of individual class accuracy) is used to resolve the tie. Further ties are solved using training accuracy.

5. Experimentation

A. Data description

The measurements used in this study were collected at the meteorological station of the University of Jaén, Andalucía (southern Spain), at coordinates 37.7877°N and 3.7782°W, and 454 meters above mean sea level. A total of 717 TSI images have been used for this study belonging to different clouds type. Concretely, 11 categories are used: Clear-Sky, Cirrocumulus, Altocumulus, Cirrus, Cirrostratus, Cumulus, Altostratus, Stratus, Stratocumulus, Nimbostratus and, Multicolor. The last category represents images with different cloud types in the same image.

The data correspond to a total of 131 days of the years 2013, 2014 and 2015 and they are a representative sample, with different solar zenithal angles of the 11 categories. Every sample was meant to be representative of 5 minute intervals, i.e., images of each of the 11 categories were carefully selected to ensure that during the 5 previous minutes period exactly the same category was maintained. Firstly, the TSI images were masked in order to highlight the border, buildings and band in the images. Secondly, the images were cropped and finally, the images were projected following [13]. This procedure transforms the images from a spherical to a rectangular grid. In order to prevent for horizon distortion effects, this transformation was conducted only for zenithal angles below 65°, i.e., a 130° field of view of the camera.



Fig. 2. Examples of sky images from the dataset. (Left: Projected image; Right: Original image)

B. Experimental methodology

To test the GP capabilities of feature generation, the proposed system is tested against expert-defined features that are commonly used in the cloud-classification problem (See section 3). This comparison will be done by estimating how many Heinle features are required to reach the same accuracy as the GP features. In order to do this, first the Heinle features are ordered using ReliefF [14],[15], an attribute selection technique that sorts features by importance. Second, groups of features are presented to the LSVM, starting with the most important one. Thus, a LSVM model is trained and tested using the first feature, a second model with the first and second features, and so on. This gives the contribution of using an increasing number of Heinle features with respect to classification accuracy. Finally, the number of Heinle features that reach the same accuracy as the genetic ones is

obtained. For example, suppose that Heinle features A, B and C are first in the ranking, and A, A+B, and A+B+C get 30%, 40% and 50% accuracy, respectively. If the genetic feature scores 45% accuracy, then it will be considered that it is equivalent (in terms of accuracy) to the first 2 Heinle features plus 0.5 of the third feature (i.e. 2.5 features).

In order to evaluate the models, a stratified 10-fold crossvalidation is used. The dataset is divided into 10 folds, each with the same number of instances per class and ordered by time. Thus, nine out of the ten folds are used to run GP, compute the Heinle features, and train the LSVM models. The remaining fold is used to test the LSVM models. This process is repeated ten times, changing the fold that is used for testing, and the average over the 10 test folds is computed.

C. Experimental results

Table II shows the 10-fold cross-validation accuracy and macro-average accuracy results of the experiments. Each column displays the results of the LSVM models constructed with the n first Heinle features (ranked by ReliefF, as explained in section 5.2). According to the methodology, new features are added to the LSVM model in order. So for instance, the result in column 2 uses the two top Heinle

features, and so on. The "GP" column shows the accuracy and macro-accuracy of LSVM using as input the genetic features. The "Replaced" column counts how many Heinle features are needed to reach the same accuracy (or macroaverage accuracy) as GP, as explained in section 5.2.

In Table II it is shown that at least four Heinle features are needed to reach GP accuracy (and macro-average accuracy). It is also observed that the standard deviation of models constructed with Heinle features is always higher than the deviation of the model with the genetic feature as input. This means that the genetic feature is more systematic, independently of the training data folds used for constructing it. Table III displays the average position taken by the feature in the ranking of Heinle features using ReliefF. For instance, μ^r has an average ranking of 3.2 because it is the third most important feature in 8 out of 10 folds, and the fourth most important in 2 folds. It can be seen that D^{rb} and C are very reliable, ranking always first and second, respectively. In Table II it was shown that more than 4 Heinle attributes are needed to obtain the same results as GP. Those four Heinle features are D^{rb} , C, μ^r , and D^{rg} (the red-blue difference, coverage, mean red, and red-green difference) because they are at the top in all 10 folds.

Table II Average and standard deviation of	f 10-fold cross-validation accura	cv and macro-average accurate	cv for the Heinle and GP features.
			· · · · · · · · · · · · · · · · · · ·

	# of Heinle Features	1	2	3	4	5	6	7	
ACCURACY	average	40.0%	45.7%	46.2%	47.8%	50.8%	59.2%	57.4%	
	std	7.2%	5.7%	6.4%	5.4%	12.6%	7.6%	7.0%	
MACRO-	average	29.6%	35.2%	37.1%	38.6%	43.0%	52.4%	50.3%	
AVERAGE	std	5.8%	4.6%	6.2%	5.8%	12.8%	6.6%	6.4%	
	#	8	9	10	11	12	GP	REPLACED	
	average	57.6%	57.8%	62.2%	62.8%	62.8%	49.0%	4.4	
ACCURACI	std	9.1%	8.9%	8.4%	6.5%	8.0%	3.2%	4.4	
MACRO-	average	50.8%	51.0%	54.4%	54.2%	54.7%	39.8%	13	
AVERAGE	std	8.3%	8.6%	7.9%	7.0%	8.1%	3.1%	4.3	

Table III. - Average ranking of Heinle features.

D ^{rb}	С	$\mu^{\rm r}$	\mathbf{D}^{rg}	\mathbf{D}^{gb}	$\operatorname{HOM}^{\mathfrak{b}}$	CON^{b}	EN^b	ENT^{b}	μ^{b}	$\boldsymbol{\gamma}^{b}$	σ^{b}
1	2	3,2	3,8	5,3	5,7	7	8	9,7	10,1	10,2	12

6. Conclusions

In this article a GP method for automatic feature extraction in the domain of cloud classification is proposed and tested. The validation against commonly used features has proven that GP is able to reach the same accuracy results as the four most important standard features.

This result is considered positive for two main reasons. The first one being that the feature generation is automatic, meaning that no expert is required in the process. The second reason is that seem GP is able to extract information from the images using fewer features than the standard set.

However, the genetic feature was unable to achieve the best accuracy and macro average possible using the 12 Heinle features. This may happen because textural features cannot be extracted with the formulation used with GP in this work. Thus, future work should explore other formulations for GP where textural and other kind of features can be evolved.

Acknowledgement

This work has been funded by the Spanish Ministry of Science under project ENE2014-56126-C2-2-R (AOPRIN-SOL project).

References

- Heinemann, D., Lorenz, E., & Girodo, M. (2006). Forecasting of solar radiation. Solar energy resource management for electricity generation from local level to global scale. Nova Science Publishers, New York.
- [2] Pankiewicz, G. S. (1995). Pattern recognition techniques for the identification of cloud and cloud systems. Meteorological Applications, 2(3), 257-271.
- [3] Pfister, G., McKenzie, R. L., Liley, J. B., Thomas, A., Forgan, B. W., & Long, C. N. (2003). Cloud coverage based on all-sky imaging and its impact on surface solar irradiance. Journal of Applied Meteorology, 42(10), 1421-1434.
- [4] Calbo J., and J. Sabburg, 2008. Feature extraction from wholesky ground-bases images for cloud-type recognition," J. Atmos. Ocean. Technol., 25(1): 3–14, 2008.
- [5] Heinle, A., A. Macke, A. Srivastav. 2010. Automatic cloud classification of whole sky images Atmos. Meas. Tech., 3 (2010), pp. 557–567
- [6] Kazantzidis, A., Tzoumanikas, P., Bais, A. F., Fotopoulos, S., & Economou, G. (2012). Cloud detection and classification with the use of whole-sky ground-based images. Atmospheric Research, 113, 80-88.
- [7] Taravat, A., Fabio Del Frate, Cristina Cornaro, and Stefania Vergari. 2015. Neural Networks and Support Vector Machine Algorithms for Automatic Cloud Classification of Whole-Sky Ground-Based Images. IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, VOL. 12(3), MARCH 2015.
- [8] Koza, J. R. (1992). Genetic programming: on the programming of computers by means of natural selection (Vol. 1). MIT press.
- [9] Tackett, W. A. (1993, June). Genetic Programming for Feature Discovery and Image Discrimination. In ICGA (pp. 303-311).
- [10] Krawiec, K. (2002). Genetic programming-based construction of features for machine learning and knowledge discovery tasks. Genetic Programming and Evolvable Machines, 3(4), 329-343.
- [11] Guo, H., Jack, L. B., & Nandi, A. K. (2005). Feature generation using genetic programming with application to fault classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 35(1), 89-99.
- [12] Back, T. (1996). Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford university press.
- [13] Marquez, R., & Coimbra, C. F. (2013). Intra-hour DNI forecasting based on cloud tracking image analysis. Solar Energy, 91, 327-336.
- [14] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In AAAI, pages 129-134, 1992.
- [15] Igor Kononenko, Edvard _Simec, and Marko Robnik-_Sikonja. Overcoming the myopia of inductive learning algorithms with relieff. Applied Intelligence, 7(1):39-55, 1997.