[1447]

OIL SPILL FEATURE SELECTION AND CLASSIFICATION USING SAR DATA

Kaiyoumu Zulipiye[1] , Fusun Balik Sanli[2]

ABSTRACT

Oil release into the ocean may affect marine ecosystems and cause environmental pollution. Thus, oil spill detection and identification becomes critical important. Characterized by synoptic view over large regions, remote sensing has been proved to be a reliable tool for oil spill detection. Synthetic Aperture Radar (SAR) imagery show returned signal that clearly distinguish oil from oil-free surface under optimal wind conditions, which makes it the most frequent used remote sensing technique in oil spill detection, but there is also have a number of oceanographic and atmospheric phenomena that also shows dark signature at SAR images will easily be mistaken with oil spill signatures.

In this study five diferent classifier (Logistic regression, Linear Bayes Normal Classifier, Quadratic Bayes Normal Classifier, K-Nearest Neighbor Classifier, Przen Windows) provided by matlab PRTolls will be tried on TerraSAR data and the results will be compared. First of all feature set will be defined by applying two different feature selection methods, namely Forward Feature Selection (FFS) and the Backward Feature Selection. Then selected classification methods will be applied. At the end the process chain will be evaluated according to the OA-test results, and the comparison between the BFS and FFS feature selection methods for each classifier will be done.

Keywords: SAR, Oil Spill, Feature Selction, Classification

1.INTRODUCTION

Synthetic Aperture Radar (SAR) images are extensively used for the detection of oil spills in the marine environment, as they are independent of sun light and not affected by cloudiness [1]. There is a number of oceanographic and atmospheric phenomena that gives rise to dark signatures in SAR images that may very easily be mistaken with oil spill signatures named lookalikes. One important aspect of lookalikes is that they are normally local phenomena, related to local features like existing currents or orography. They are often also seasonal, like algae blooms, or ice [2][3].

İn this study oil spill faeture selection and classification methods are will presentented.

Researchers have used different input features for oil spill classification in their studies. Several studies indicate this notice. Fiscella used 14 features [10], Solberg and Theophilopoulos used 15 features[11], Solberg used 11 features [12], many of which were different from the previous studies. A general description about the calculated features is given by Espedal and Johannessen[13], in which texture features are introduced for the first time. Moreover, Keramitsoglou [14] refer to 14 features and Karathanassi [15] use 13 features covering physical, geometrical and textural behavior.

Several studies try to unify all the features used having similar characteristics [16], [17].

Konstantinos Topouzelis, Apostolos Psyllos [1] used 25 most commonly used features in the scientific community was examined.

The absence of a systematic research on the extracted features as well as their contribution to the classification results, forces researchers to arbitrarily select features as inputs to their systems. Previous research [18], [19] headed, for the first time, on this direction.

2.DATA AND METHODOLOGY

In this study we used high resolution terrasar_x oil spill sample data provided by NIK company and some lookalikes terrasar_x data includes several sea states. From the 17 terrasar_x images 120 image windows are extracted containing 60 oil spill and 60 lookalikes and 34 features is extracted.

Our goal in this study is to leverage a set of n training samples in order to design a classifier that is capable of distinguishing between m classes on the basis of an input vector x, where x = [x1, ..., xd]T ∈ Rd are simply the d dark patch features. By using the “Leave-One-Out” method, we remove one item at a time from the n + 1 element dataset and use the remaining n elements for training the classifier. In our case the dataset contains 120 elements (n

= 120), we have just two classes (m = 2), and d =34. We adopt the common technique of representing the class labels using a "1-of-m" encoding vector y = [y(1), y(2), ..., y(m)] such that yi = 1 if x corresponds to an example belonging to class i and yi = 0 otherwise. The n training samples that we use in each cycle of the "Leave-One-Out" can thus be represented as a set of training data D = [(x1, y1), ..., (xn, yn)]. In each cycle of the "Leave-One-Out", after training the classifier using the data D, we test the classifier in the remaining item. The classification results are the mean values of all performed tests.[4][5]

In this work we have exploited an approach like: apply a feature selection methods first and then test the different classifiers using the selection features. The feature selection step is important because many of the features we have computed are redundant or strong correlated. On the other hand, due to the limited number of samples in our database, using too many features would ultimately cause a decrease of performance due to overfitting. Feature selection methods essentially divide into wrappers, filters, and embedded methods. Wrappers utilize the learning algorithm of interest as a black box to score subsets of variable according to their predictive power. Filters select subsets of variables as a pre-processing step, independently of the chosen classifier. Embedded methods perform variable selection in the process of training and are usually specific to given learning algorithm [6]. Following these definitions we can see that in this study approach uses a filter type selection. We have tested two different suboptimal feature selection algorithms: the Forward Feature Selection (FES) [6] method and the Backward Feature Selection (BFS) [6] method. Depending on the application and the objectives they may lead to different subsets. One approach may be preferred over the other one. In particular backward elimination procedures may yield better performances but at the expense of possibly larger feature sets. However if the feature set is reduced too much, the performance may degrade abruptly [7].

In this work, we have used the matlab toolbox PRTools (see http://www.prtools.org/), that provides FFS and BFS routines. The evaluation criteria was the sum of the Mahalanobis distances. After the feature selection, we used standard classifiers provided also by PRTools. We have tested the following:

Logistic regression (LOGC)
Linear Bayes Normal Classifier (LDC)
Quadratic Bayes Normal Classifier (QDC)
K-Nearest Neighbor Classifier (KNNC)
Przen Windows PARZENC()

Logistic regression is adequate when we have a dependent variable y which takes binary values, which is also our case in this work (1:= oil, 0:= lookalike). We then assume that y follows a binomial distribution (y ∼ Bin (p, n)), where p is the probability of success (being oil) and n is the number of trials (number of dark patches/samples). The mean value of y is np, and if we normalize the variable it is simply p, taking values between 0 and 1. The model in logistic regression proposes that we use as dependent variable the logarithm of the odds ratio (so-called logits). Odds are defined as Odd := p/(1 − p) and logits are given by expression

$$\mathrm{logit}(p) = \ln\left(\frac{p}{1-p}\right),$$

and they are modeled as a linear function of the zi, the set of explanatory variables (the features), that might inform the final probability:

$$\mathrm{logit}(p) = a + \beta z$$

where a is called the intercept, β = (β1, β2, . . .) are called the regression coefficients, z = (z1, . . .) is the vector of the features zi and where p = Pr(y = 1|z). The logit is then converted into a probability using the following expression (the logistic function):

$$p = \frac{e^{\mathrm{logit}}}{1 + e^{\mathrm{logit}}}.$$

The logistic model automatically assures that p is bounded by 0 and 1, which is not the case with normal linear regression. The parameters of the model are a and β and can be obtained for example using Ridge Regression. We used the implementation of the PRTools. An interpretation of the coefficients, in the case of a dichotomous explanatory variable (zj is 0 or 1), is that odds for group with zj = 1 are exp(βj) higher, other parameters being equal. The classifiers used in this work referred to as LDC and QDC are common used Bayesian classifiers. They are based on the prerequisites that we can quantify as a cost the "damage" involved when an object is wrongly classified. A second prerequisite, when using Bayesian classifiers, is that the expectation of this cost can be used as an optimization criteria. The cost function is in our case, where we have two classes, a 2x2 matrix C(ˆ c, ck), where c is the class and can take the values c1 and c2. It can be demonstrated [8] that the expectation of the cost that the classifier assigns a class ˆ ci to a measurement vector z (containing the features) which actually corresponds to an object with true class ck, is

$$\mathrm{R}(\hat{c}_i|z) = \sum_{k=1}^{K} \mathrm{C}(\hat{c}_i|c_k)\mathrm{P}(c_k|z)$$

where K is the number of classes and P(ck|z) is the posterior probability. This expression is called the conditional risk and by averaging it over all possible all possible measurements we obtain the overall risk, which we would like to minimize in this approach. As a conclusion, a Bayesian classifier assigns a class ˆ cBayes (z) to a measurement vector z, such that

$$\hat{c}_{Bayes}(z) = \mathrm{argmin}_{c\in\ \mathcal{L}}\left\{\mathrm{R}(c|z)\right\}$$

The linear and quadratic classifiers used result from further assumptions:
a uniform cost function is defined, where a unit cost is assumed when an object is misclassified, and zero cost when the classification is correct;

the conditional probability densities p(z|ck) are modeled as Gaussian functions. With the above two assumptions, the classifier is called a quadratic classifier and if we make a further simplification that the covariance matrices in the Gaussian functions are class independent the classifier is called linear.

The K-Nearest Neighbor algorithm is the simplest approach for two-class classification. It is a non-parametric model, which classifies a sample by assigning it the label most frequently represented among the K nearest samples. It uses directly the training set without explicitly estimating probability densities. For classifying a new sample, a distance function is needed to determine which K members of the training set are closest to it. Once the K-nearest training instances have been found, their class assignments are used to predict the class for the new instance, by a majority vote. It can be shown that the performance of the K-NNR approximates the optimum as K increases, but this asymptotic optimality only holds true if the training set is dense. In practice, the demand on the size of the training set is very high, implying increased computational complexity. A suitable choice is to make K proportional to √Nk, where Nk is the number of samples belonging to class k [8].

Parzen windows classification is a technique for nonparametric density estimation, which can also be used for classification. It can be regarded as a generalization of the k-nearest neighbor technique. The basic idea is that the knowledge gained by the observation of sample zi is represented by a function h centered at zi (the so-called kernel of the estimator) and with an influence restricted to a small vicinity. This function represents the contribution of the observation data zi to the estimative of the conditional probability p (z|c = ck). For obtaining the final estimate, all contributions for all training data are summed together [8].

Please refer to [82] for more details on the classifiers and to the documentation of

PRTools [9] for details on the implementation.

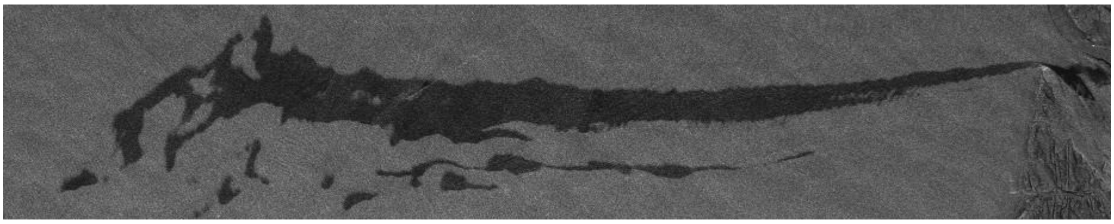
A total of 34 features have been considered for classification. These features have all been extracted from the segmented dark patch, The used features can be divided in four main groups: geometrical, backscatter, texture features [5]

Geometrical Features	Area (A)
	Perimeter (P)
	Complexity (C)
	Length (L)
	Width (W)
	Length To Width Ratio (LWR)
	Compactness (Comp)
	First Invariant Planar Moment (FIPM)
	Ellipse-Length (EL)
	Ellipse-Width (EW)
	Ellipse-Asymetry (EA)
	Form Factor (FF)
	Spreading (S)
Backscatter Features	Inside Slick Radar Backcatter (μ_{obj})
	Inside Slick Standard Deviation (σ_{obj})
	Outside Slick Radar Backscatter (μ_{sce})
	Outside Slick Standard Deviation (σ_{sce})
	Intensity Ratio
	Intensity Standard Deviation Ratio
	Intensity Standard Deviation Ratio Inside (ISRI)
	Intensity Standard Deviation Ratio Outside (ISRO)
	ISRI ISRO Ratio
	Min Slick Value (MinObj)
	Max Slick Value (MaxObj)
	Max Contrast (ConMax)
	Mean Contrast (ConMe)
	Max Gradient (GMax)
	Mean Gradient (GMe)
	Gradient Standard Deviation (GSd)
Texture Features:	GLCM Homogeneity
	GLCM Contrast
	GLCM Entropy
	GLCM Correlation
	GLCM Dissimilarity

In order to provide a performance measure of the classifier, which is then also considered by extension to be the performance of the whole detection system, normally two Overall Accuracies (OA) are used in the literature: the oil spill detection rate (OA-oil) and the lookalike detection rate (OA-lookalike). Occasionally, the total accuracy (OA-test) is also used.

3.EXPERIMENTAL RESULTS

One interesting example is the " Length To Width Ratio" (LWR), as many of the detected oil spills originate from discharges of moving vessels, presenting a linear shape. In the empirical guidelines provided to oil spill detection operators, linear shape is considered a strong indication of oil spill. Figure 3.1 depicts the histogram of the LWR values calculated for the data set used in the classification.



3.1 Example of oil slick from ongoing discharge of moving vessel, with LWR=164

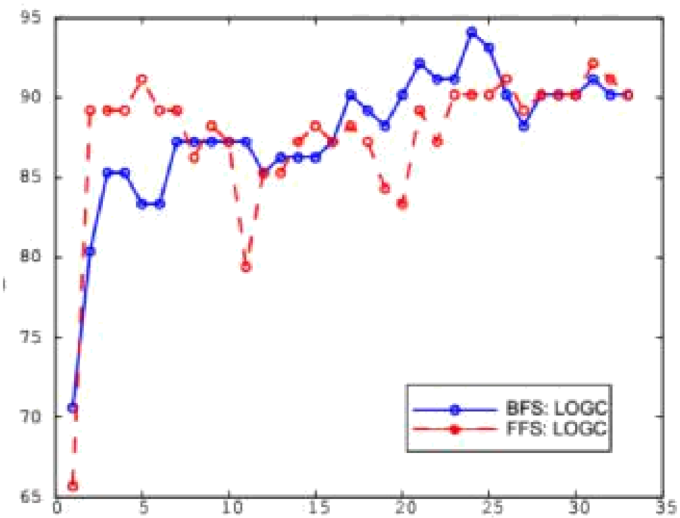
We have run the feature selection methods BFS and FFS varying the selected feature set size from 1 to the maximum value 34 and applied a number of different classifiers as described.Table 3.1 provides an overview of the obtained results when no wind information is used. For the two feature selection methods and for each classifier, the maximum obtained OA-test and the correspondent respective number of selected features are again given. For LOGC and LDC classifiers, the results can be visualized in more detail in Figure 3.2 and Figure 3.3

Classifier	BFS	FFS
LOGC	OA-test = 94.1 NrFeat = 24	OA-test = 92.2 NrFeat = 31
LDC	OA-test = 91.2 NrFeat = 18	OA-test = 91.2 NrFeat = 29
QDC	OA-test = 93.1 NrFeat = 14	OA-test = 89.2 NrFeat = 19
KNNC	OA-test = 91.2 NrFeat = 31	OA-test = 92.1 NrFeat = 20
PARZENC	OA-test = 90.2 NrFeat = 11	OA-test = 91.2 NrFeat = 5

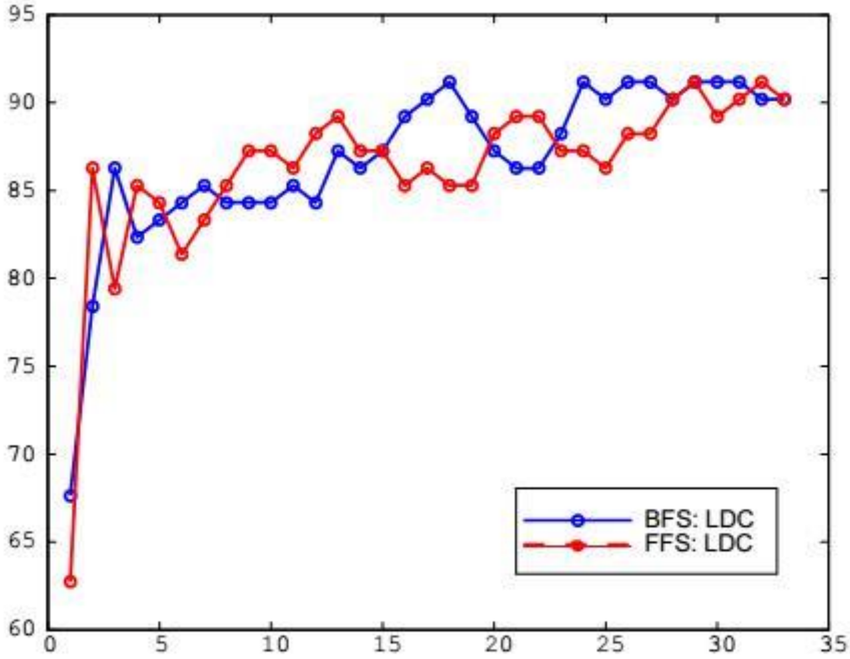
3.1. Classification Results for each classifiers

4.CONCLUSION AND DISCUSSION

We have tested an approaches for automatic classification of dark patches in terraSAR_x images. For training and testing our algorithms we have first built a 120 dark patch database containing oil spills and lookalikes detected by experienced operators. In this approaches we assessed the problem of feature selection: by adopting standard feature selection methods, namely the BFS and the FFS, and then applying standard classifiers.



3.2 Evolution of OA-test with the number of selected features for LOGC classifier, Comparison between the BSF and FFS feature selection methods (X represent number of selected features， Y represent OA test results)



3.3 Evolution of OA-test with the number of selected features for LDC classifier, Comparison between the BSF and FFS feature selection methods (X represent number of selected features， Y represent OA test results)

This study shows best results, with the high accuracy corresponding to the classifiers such as LDC and LOGC methods using different feature selection methods.

We believe there is a clear evidence that automatic classification could be an option to the study of oil spill detection. Although we believe to have obtained very good results when compared to the other classification applications we recognize the limitation dueto the reduced size of our database, so this study could be developed and test with another big data sets to promotion of the automatic methods of classification. Because of the lack of data sources we did not use the wind information as a feature in this study but as all knows the wind information is important in the oil spill detections , so the methods applied in this study could tested with wind information as further study.

REFERENCES

Konstantinos Topouzelis, Apostolos Psyllos. *Oil spill feature selection and classification using decision tree forest on SAR image data* 2012.

SAR Ocean Feature Catalogue, ESA, October 1994. 1, 4, 85

W. Alpers, "The tropical and subtropical ocean viewed by ers sar," last modified

(2009), available at http://earth.esa.int/applications/ERS-SARtropical

M. Figueiredo B. Krishnapuram, L. Carin and A. Hartemink, "Sparse multinomial logistic regression: fast algorithms and generalization bounds," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 6, June 2005. 17, 88, 91, 96

S´onia Antunes Pelizzari "*Oil Spill Detection Using SAR Images*" 90,91,92

I. Guyon and A. Elisseeff, "An introduction to variable and feature selection,"

Journal of Machine Learning Research, vol. 3, pp. 1157–1182, 2003. 91

I. Guyon et al., Feature Extraction: Foundations and Applications, Springer, 2006. 46, 92

F. van der Heijden et al., Classification, Parameter Estimation and State Estimation, Wiley, West Sussex, 2004. 93, 94, 95

R. Duin et al., "Prtools4, a matlab toolbox for pattern recognition," available at http://http://www.prtools.org/files/PRTools4.1.pdf, 2007. 95

Fiscella, B., Giancaspro, A., Nirchio, F., Trivero, P., 2000. Oil spill detection using marine SAR images. International Journal of Remote Sensing 21 (18), 3561–3566.

Solberg, R., Theophilopoulos, N.A., 1997. Envisys - A solution for Automatic oil spill detection in the Mediterranean, In: Proceedings of the Fourth Thematic Conference on Remote Sensing for Marine and Coastal Environmentsvol. 1, Environmental Research Institute of Michigan, Ann Arbor, Michigan, 3–12.

Solberg, A., Storvik, G., Solberg, R., Volden, E., 1999. Automatic Detection of Oil Spills in ERS SAR Images. IEEE Transactions on Geoscience and Remote Sensing 37 (4),1916–1924.

Espedal, A., Johannessen, A., 2000. Detection of oil spills near offshore installations using synthetic aperture radar (SAR). International Journal of Remote Sensing 21 (11), 2141–2144

Keramitsoglou, I., Cartalis, C., Kiranoudis, C., 2005. Automatic identification of oil spills on satellite images. Environmental Modeling and Software 21 (5), 640–652.

Karathanassi, V., Topouzelis, K., Pavlakis, P., Rokos, D., 2006. An object-oriented methodology to detect oil spills. International Journal of Remote Sensing 27 (23), 5235–5251.

Brekke, C., Solberg, A., 2005. Oil spill detection by satellite remote sensing. Remote sensing of Environment 95 (1), 1–13.

28.Montali, A., Giancito, G., Migliaccio, M., Gambardella, A., 2006. Supervised pattern classification techniques for oil spill classification in SAR images: preliminary results, SEASAR 2006 Workshop, ESA-ESRIN, Frascati, Italy, 23–26 January

29.Stathakis, D., Topouzelis, K., Karathanassi, V., 2006. Large-scale feature selection using evolved neural networks, In Proceedings of SPIE, Image and Signal Processing for Remote Sensing XII, Bruzzone (Ed.), 6365.

30.Topouzelis, K., Stathakis, D., Karathanassi, V., 2009. Investigation of genetic algorithms contribution to feature selection for oil spill detection. International Journal of Remote Sensing 30 (3), 611–625.