

Reporting on Three Years of Activity on Oil Spill Classification from Optical Satellite Images

Linda Corucci^a, Marco Cococcioni^b, Andrea Masini^c

^a Dipartimento di Ingegneria dell'Informazione: Elettronica, Informatica, Telecomunicazioni. University of Pisa

Largo Lucio Lazzarino, 1– 56122 Pisa, Italy

linda.corucci@iet.unipi.it

^b NURC - a Nato Research Centre

m.cococcioni@ieee.org

^c Flyby S.r.l.

Via Puini, 97, 57128 Livorno, Italy

andrea.masini@flyby.it

Introduction

Objective: to investigate the potential of oil spill classification from optical satellite images

- We studied oil spills by exploiting a large dataset of oil spills detected from optical satellite sensors
- We first employed simple statistical classifiers and neural networks and we measured the performance by means of the correct classification percentage
- We then moved to ROC analysis, using a more interpretable classifier (ANFIS)
- Finally, we have looked for a technique which could be suitable for:
 - Online classification
 - Cost-oriented classification



We have developed a technique based on an ensemble of cost-oriented, incremental and decremental SVMs, exploiting the concept of the ROC Convex Hull and of concavities repair.

Oil spill detection: problem description

The environmental damage caused by the **creation of spills of hydrocarbon compounds over the sea surface** represents a relevant issue of increasing public concern.

Causes:

- Oil-tanker accidents
- Illegal cleaning of tankers

- Need for an efficient and cost effective monitoring system allowing for coast guard and environmental protection authorities alert

- False alarms should be avoided as much as possible

The development of a new approach based on optical data could be used either on its own or as a support to SAR-based solutions.

Multi-spectral satellite images exploitation can efficiently improve oil pollution detection and monitoring:

- Large areas can be monitored in an economical and easy way
- Images of the sea surface are regularly available, including remote areas

Operational approach:

- 1) Oil slick candidate selection in the satellite image
- 2) Physical and geometrical features extraction  object characterization
- 3) Object classification: oil spill or look-alike

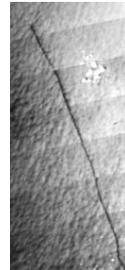
The dataset

We have built a dataset of regions of interest by collecting a number of optical images taken during the **years 2008 and 2009** over the entire area of the **Mediterranean Sea**.

- **MODIS-TERRA** and **MODIS-AQUA L1B data** @ 250 m spatial resolution
 - b1: VIS @ 0.65 μm
 - b2: NIR @ 0.85 μm
- **clear sky conditions**
- Oil slicks were detected by **visual inspection** performed by a **trained interpreter** basing on his experience . Region of interest were manually selected
- The selected regions have been divided into two classes:

Oil spills:

regions identified by photointerpretation as possible oil spills on the basis of the contrast between the regions and the surrounding areas (which are supposed to be clean waters).



Look-alikes:

which could be algal blooms, currents etc...



- For each selected image the **corresponding Envisat-ASAR image** @ 150 m spatial resolution has been analyzed for comparison.
- For many of the selected cases **ground truth** was available

The number of oil slicks detected is approximately equal to the number of look-alikes for a total 304 elements: 157 oil spills and 147 look-alikes

Feature description

In order to discriminate between oil spills and look-alikes, we exploited a number of physical and geometrical features characterizing the object to be classified.

Geometrical features

- *Area of the object*
- *Perimeter*
- *Complexity*: $C = P / 2\sqrt{\pi A}$ → Small values for regions with simple geometry, larger values for complex geometrical regions.
- *Spreading*: computed performing a principal component analysis on the vector whose components are the coordinates of the pixels belonging to the object:

$$S = 100 \lambda_2 / (\lambda_1 + \lambda_2)$$

eigenvalues associated to the covariance matrix ($\lambda_1 > \lambda_2$)

Low values for long and thin objects. High values for objects close to a circular shape

Grey level features → calculated on the ratio: band2/band1

- *Object Standard Deviation*: st. dev. of the intensity of pixels belonging to the object.
- *Max. Contrast*: difference between the background intensity mean value and the lowest intensity value inside the object.
- *Mean Contrast*: difference between the background intensity mean value and the object intensity mean value.

Batch classification approach: first attempt

As a first attempt to oil spill classification, we applied a set of statistical classifiers and neural networks to the extracted features.

- Training set: 60% randomly chosen examples of the available dataset
- Test set: the remaining 40%

Classifiers

- Normal density-based linear classifier
- quadratic classifier
- logistic regression-based linear classifier
- k -nearest neighbor classifiers ($k=1,2,3$)
- radial basis function (RBF) neural network
- multi layer perceptron (MLP) neural network

Results

| Classifier | Mean training set correct classification % | Mean test set correct classification % |
|--------------------------------|--|--|
| Normal density based linear | 79 | 74 |
| Normal density based quadratic | 77 | 75 |
| Logistic regression | 80 | 74 |
| 1-nearest neighbor | 100 | 71 |
| 2-nearest neighbor | 92 | 75 |
| 3-nearest neighbor | 88 | 76 |
| RBF (10 neurons) | 76 | 65 |
| MLP (6 hidden neurons) | 92 | 79 |

The estimated relative error is 2%.

Batch classification approach: second attempt (1/2)

- Search for a more interpretable classifier (ANFIS)
- Refine performance evaluation procedure (ROC analysis)

The model

ANFIS (Adaptive Network-based Fuzzy Inference System) [1]

$$r_1 : \text{IF } X_1 \text{ is } A_{1,1} \text{ and } X_2 \text{ is } A_{2,1} \text{ THEN } y_1 = p_{1,0} + p_{1,1}X_1 + p_{1,2}X_2$$

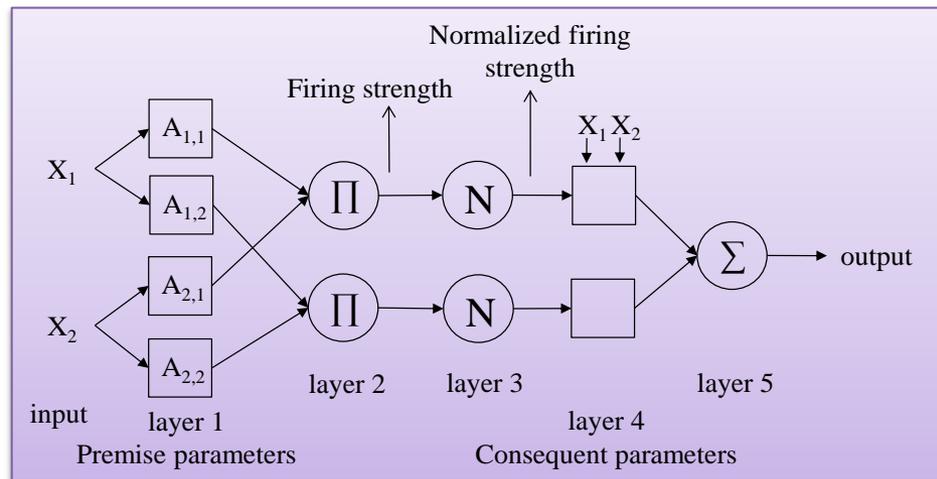
$$r_2 : \text{IF } X_1 \text{ is } A_{1,2} \text{ and } X_2 \text{ is } A_{2,2} \text{ THEN } y_2 = p_{2,0} + p_{2,1}X_1 + p_{2,2}X_2$$

Example of Takagi-Sugeno rules

The associated ANFIS network

Advantages:

- 1) Rule structure and membership functions are derived from the dataset by means of the supervised learning procedure
- 2) Interpretability is preserved
- 3) ANFIS can be regarded as a grey-box model
- 4) Only few parameters must be provided.



Hybrid learning algorithm for ANFIS training:

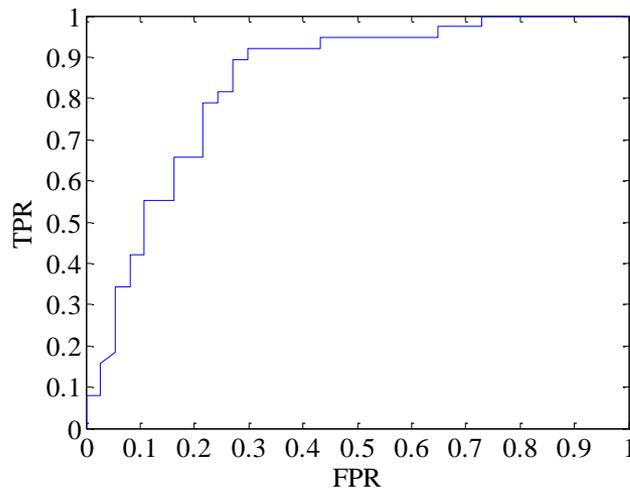
- backpropagation gradient descent method (antecedent parameters)
- least-squares method based on Kalman filter (consequent parameters)

[1] Jang, J.-S. R., "ANFIS: adaptive-network-based fuzzy inference systems," IEEE Trans. Syst. Man. Cybernet. 23(3), 665-685 (1993).

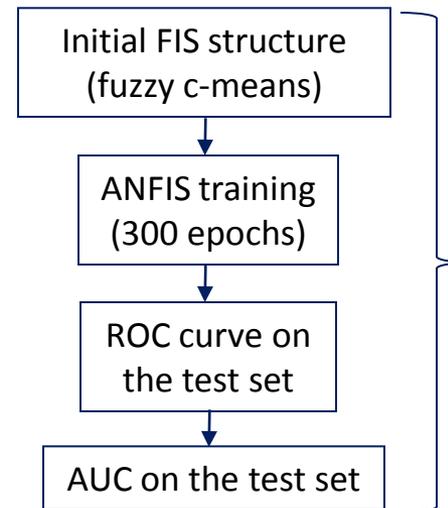
Batch classification approach: second attempt (2/2)

Results

- **Training set:** 60% randomly chosen examples of the available dataset
- **Test set:** the remaining 40% (used to evaluate the classification performance)
- **Membership functions:** Gaussian for the inputs and linear for the output
- **Initial Fuzzy Inference System structure:** data were partitioned using a fuzzy c-means clustering algorithm → number of clusters: number of rules and the membership functions for the antecedents and consequents in the generated FIS



a typical ROC curve obtained on the test set



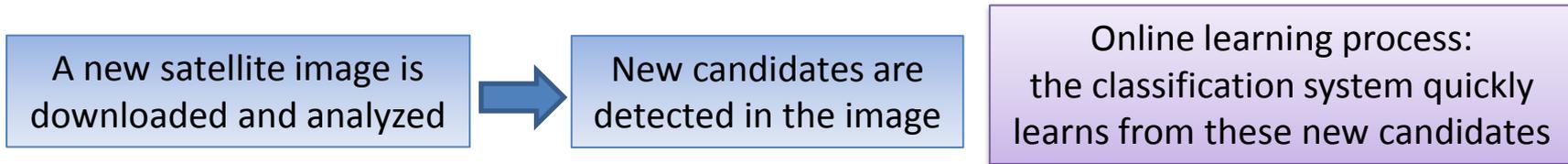
Iterative procedure:
Choice of the
number of clusters
corresponding to
the maximum mean
AUC on the test set

On the test set we achieved a mean AUC of 0.80

Towards online cost-oriented classification (1/2)

Motivations

Online learning: operative scenario → An existing dataset is updated by some new oil spill candidates



Cost-oriented classification

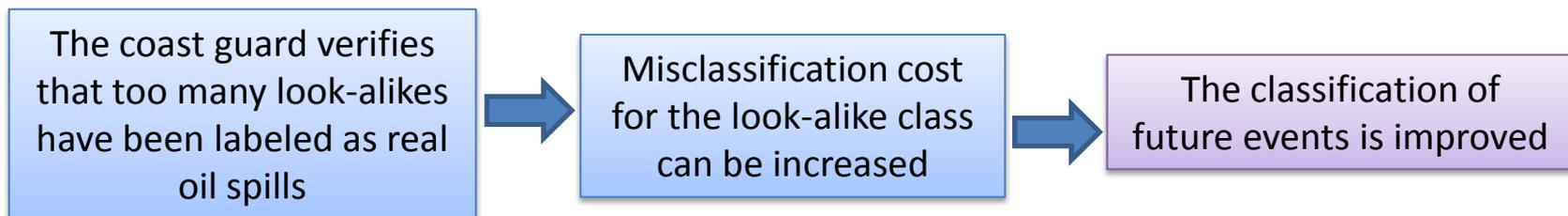
It is desirable to assign a different cost to the misclassification errors for the oil spill and for the look-alike classes: this allows to perform an optimal classification with respect to a chosen cost index for class misclassification (a *cost-oriented classification*)

Proposed approach:

- Combination between online learning and cost-oriented classification
- Use of time varying costs

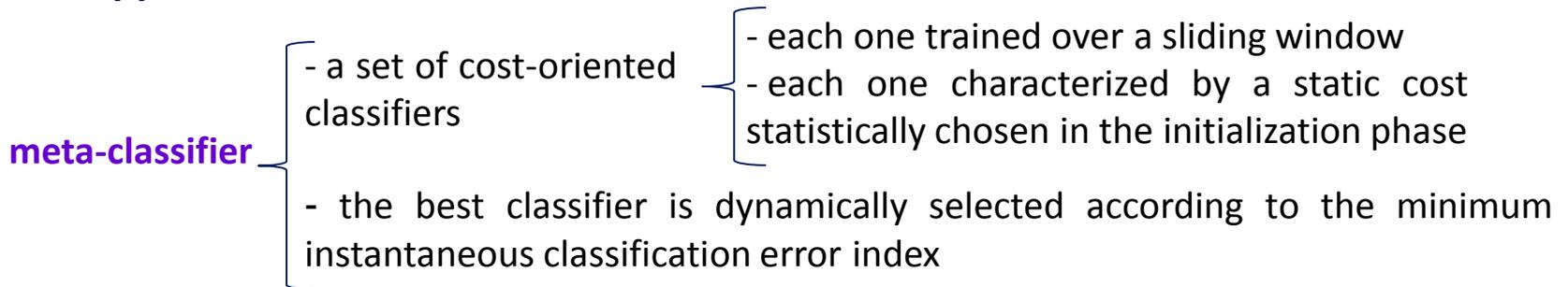
→ Desired target conditions (e.g. maximum false alarm rate) can be changed according to the latest classification results

Example: operative scenario



Towards online cost-oriented classification (2/2)

Our approach: an ensemble of cost-oriented classifiers



Using an ensemble reduces instabilities which are likely to occur when using a single cost-oriented classifier, trained over a sliding window.

Classifier type: Support Vector Machines (SVMs)

- Availability of a cost-oriented training algorithm
- Availability of an incremental and decremental training algorithm (adaptability: the system is continuously updated)

We have integrated the cost-oriented formulation of SVMs (CO-SVMs) [1] and the incremental/decremental formulation of SVMs (ID-SVMs) [2] into a unique framework named COID-SVMs [3]. Then we have used an ensemble of COID-SVMs.

Drawback: loss of interpretability

[1] Cortes, C., and Vapnik, V., "Support vector networks," Mach. Learn. 20, 273–297 (1995).

[2] Cauwenberghs, G., and Poggio, T., "Incremental and decremental support vector machine learning," Adv. Neural Information Processing Systems (NIPS*2000), Cambridge, MA: MIT Press, vol. 13 (2001).

[3] Corucci, L., Nardelli, F., Cococcioni, M., "Building a time variant cost-oriented classifier using an ensemble of SVMs on a real case Application", in Proc. of IEEE SMC 2010 Conference, Istanbul, Turkey, 10-13 October 2010.

Cost-oriented classification in ROC space (1/3)

ROC convex hull

Basing on the definition of TPR and FPR , in a two-class cost-oriented classification problem the following cost index is often used:

$$J(TPR, FPR) = FPR \cdot C(Y, \mathbf{n}) \cdot p(\mathbf{n}) + (1 - TPR) \cdot C(N, \mathbf{p}) \cdot p(\mathbf{p})$$

cost function associated to the misclassification of a \mathbf{n} pattern α -priori probability for negatives cost function associated to the misclassification of a \mathbf{p} pattern α -priori probability for positives

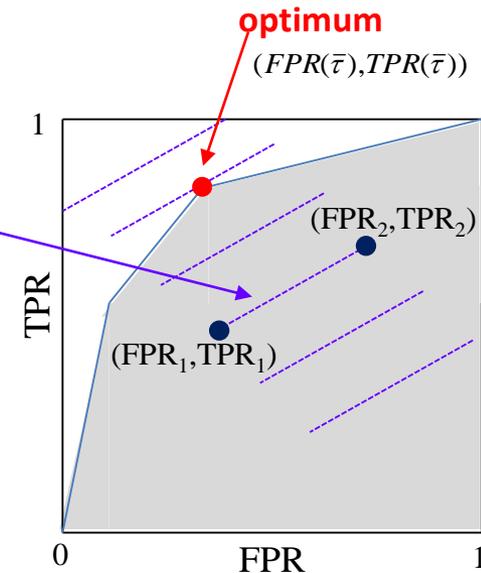
Optimal threshold for a trained classifier: $\bar{\tau} = \arg \min_{\tau_i} (J(TPR(\tau_i), FPR(\tau_i)))$.

When J is used as the cost index, *iso-performance* curves are straight lines in the ROC space [4]:

$$\frac{TPR_2 - TPR_1}{FPR_2 - FPR_1} = \frac{C(Y, \mathbf{n}) p(\mathbf{n})}{C(N, \mathbf{p}) p(\mathbf{p})}$$

slope

- A set of costs and class distributions corresponds to a family of iso-performance lines (same slope)
- Within a family of iso-performance lines, the line that is tangent to the ROC curve identifies the **optimal point**.



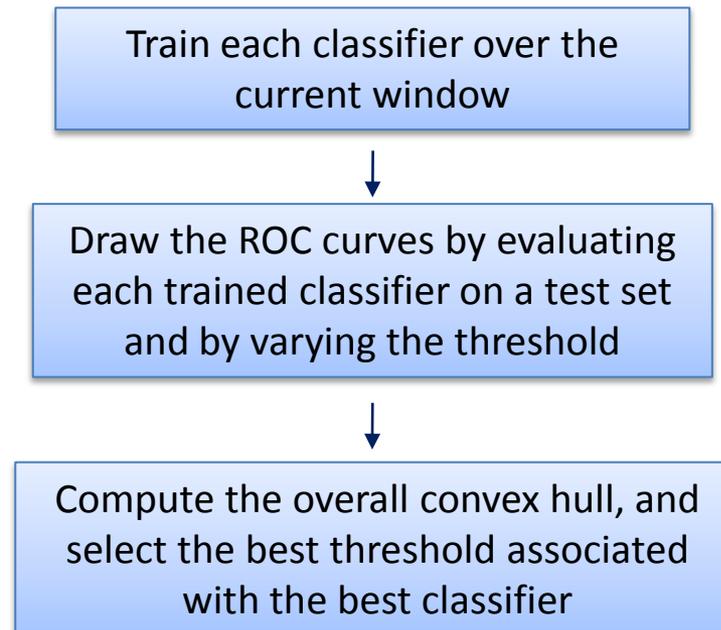
This point lays on the **ROC convex hull**

[4] Provost, F., and Fawcett, T., "Robust classification for imprecise environments," Mach. Learn. 42, 203–231 (2001).

Cost-oriented classification in ROC space (2/3)

ROC convex hull for an ensemble of classifiers

In order to perform a cost-oriented online classification based on an ensemble of continuous classifiers we adopted the following strategy:



Doing this, the optimal classifier and threshold are selected.

Cost-oriented classification in ROC space (3/3)

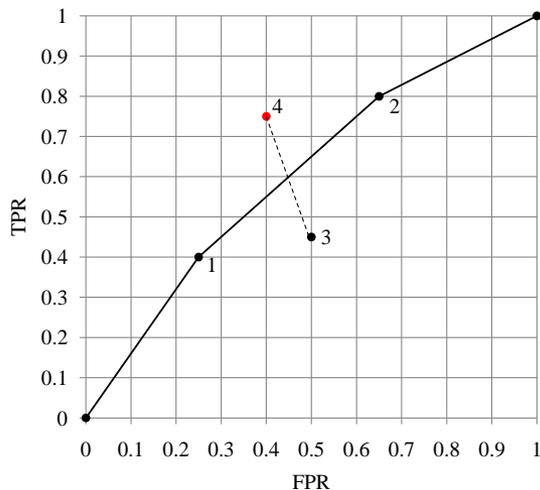
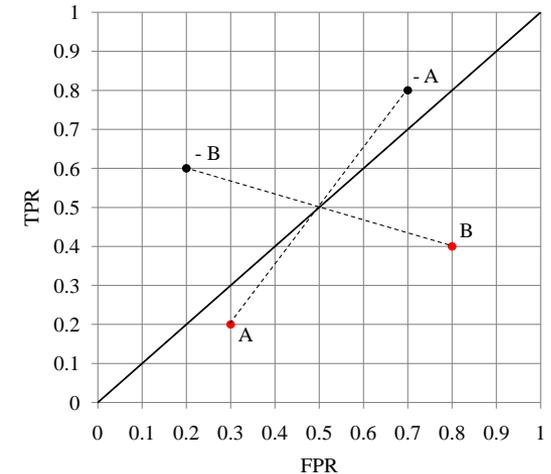
Repairing concavities in ROC curves

The performance can be improved by exploiting a technique for repairing concavities in the ROC curves [5].

Observation:

Classifiers below the ascending diagonal in the ROC space perform worse than the random classifier, thus by simply inverting their predictions, the resulting classifiers will perform better than the random one.

The *SwapOne* algorithm generalizes the prediction inversion procedure to linear segments connecting arbitrary classifiers.



The area under the convex hull can sometimes be increased by mirroring points under the convex hull to the other side. The resulting mirrored points will be located in a more northwest region of the ROC space.

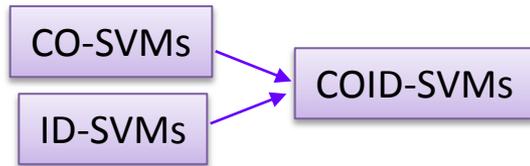
➡ **The convex hull is consequently improved**

[5] Flach, P. A., Wu, S., "Repairing concavities in ROC curves," in Proc. 2003 UK Workshop on Computational Intelligence, pp. 38–44, 2003.

Support Vector Machines

COID-SVMs for online classification

COID-SVM is a combination between CO-SVMs and ID-SVMs:



It includes:

- **Cost-oriented classification in the ROC space** by means of the ROC CH method, and CH improvement through concavities repairing
- **Online classification**: data acquisition happens incrementally

The SVM structures can be dynamically modified, instead of being fixed by the dimension of the data acquired in batch mode. This improves the **adaptability** to time varying conditions.

Online approach:

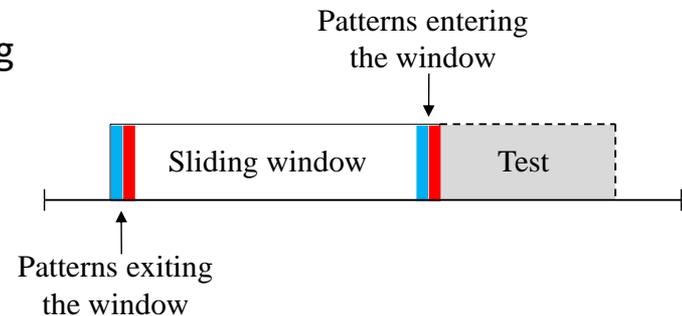
The training algorithm, which is performed over a sliding window, follows three steps:

- 1) SVM initialization
- 2) SVM incremental learning
- 3) SVM decremental unlearning

Training set: odd elements within the window

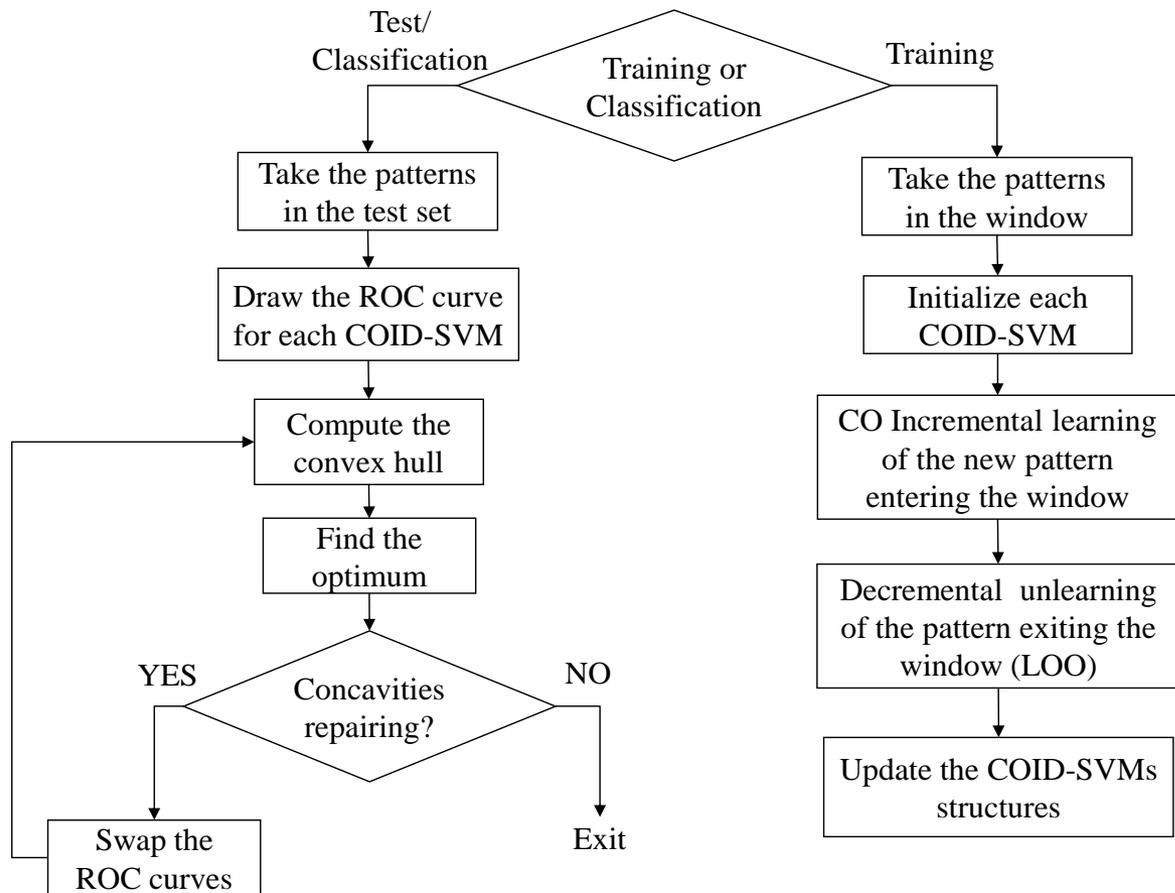
Validation set: even elements within the window (used to evaluate the expected costs, to build the ROC curves and the convex hull, and to repair concavities)

Test set: patterns following the window (size equal to the validation and training set size)



The ensemble of COID-SVMs in the ROC space

Instead of using just one COID-SVM with dynamically changing $C(Y, \mathbf{n})$ and $C(N, \mathbf{p})$, we used a set of 5 COID-SVMs with static misclassification costs: COID-SVM 1, COID-SVM 2,..., COID-SVM 5



Advantages:

- Better performance;
- More robust and stable performance;
- Static cost functions give better stability to the SVM online training;
- Both the search for the optimum and concavities repairing are performed taking into account all the classifier ensemble



Fusion of the information provided by each classifier in place of a dynamic classifier selection

A software for online cost-oriented classification

Graphic interface:

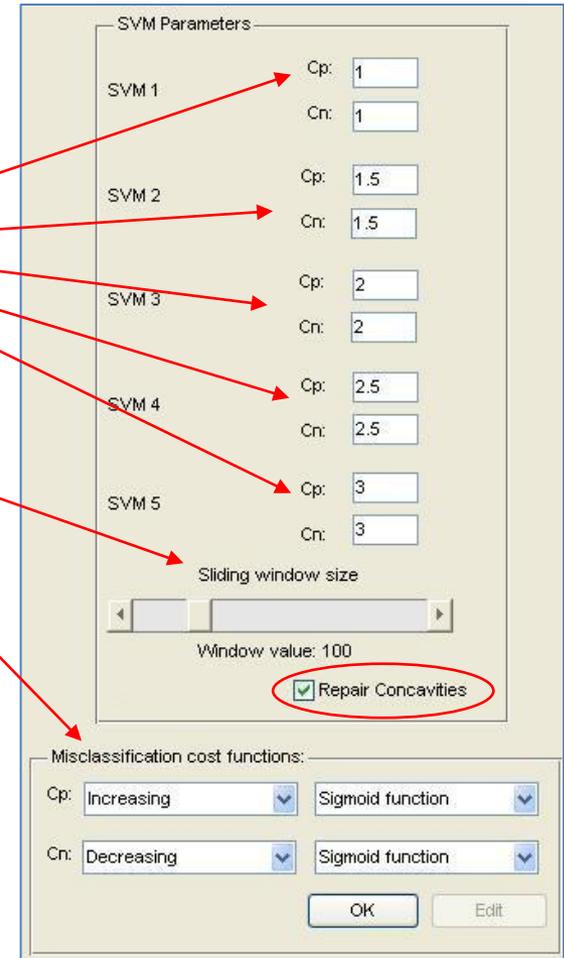
It allows the user to set the inputs and to obtain the optimum for the classification basing on the convex hull method.

- Dataset selection
- Static misclassification costs setting for each COID-SVM in the ensemble ($M(Y, \mathbf{n})$ and $M(N, \mathbf{p})$)
- Sliding window size setting
- Concavities repairing
- Cost functions setting for the iso-performance lines ($C(Y, \mathbf{n})(t)$ and $C(N, \mathbf{p})(t)$)

For each data subset entering the sliding window the software produces a plot showing:

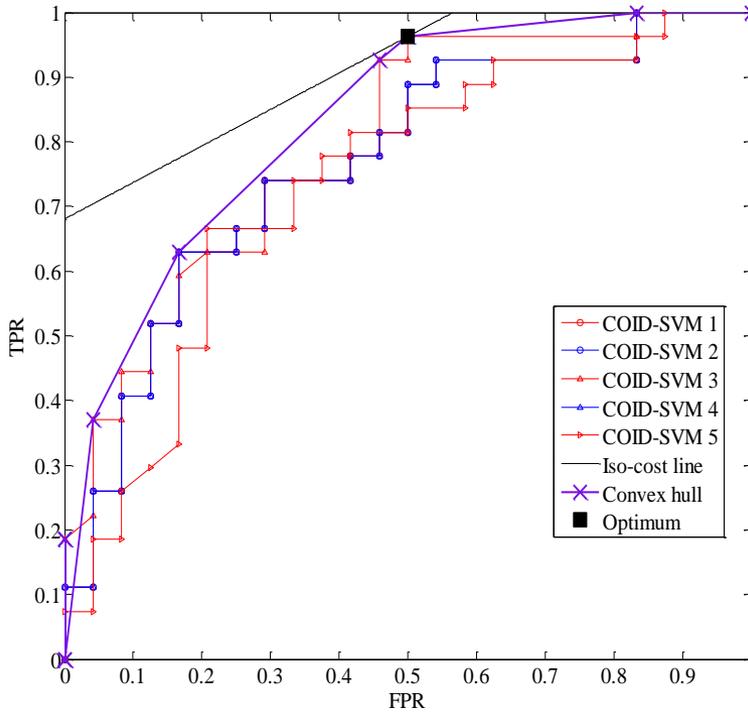
- ROC curve for each SVM
 - Convex hull
 - Optimum for the classification
- } computed over the validation set

and computes the cost index $J(TPR, FPR)$ for the ensemble on the validation set and on the test set

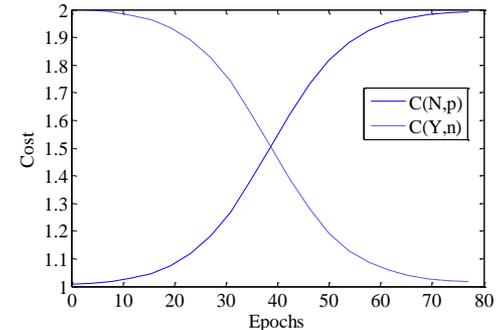


Results: Experiment without concavities repair

A typical result for an online classification step:



- Sliding window size: 100 elements (50 for training 50 for validation)
- 50 test elements used for each online epoch
- Time varying sigmoid cost functions:



- Equal static misclassification costs for each SVM

Performance analysis on the complete dataset:

| | Integrated global cost index for the ensemble applied to the dataset (77 online epochs) | Mean area under the convex hull (test set) |
|----------------------------|---|--|
| Without concavities repair | 92.89 (test set), 77.39 (val. set) | 0.70 |
| With concavities repair | 91.03 (test set, 2% decrease) , 73.52 (val. set, 5% decrease) | 0.72 (3% increase) |

Conclusions

- We have described how we tackled oil spill classification from optical satellite images by exploiting many different machine learning techniques.
- We started our analysis by employing simple statistical classifiers and neural networks measuring the performance by means of the correct classification percentage.
- Taking into account the limitations of performance evaluation based on classification accuracy in a two-class problem, we moved to ROC analysis, using a more interpretable classifier (ANFIS)
- Finally, since oil spill dataset is collected incrementally, we have looked for a technique which could be suitable for:
 - online classification
 - cost-oriented classification
- We have shown how using an ensemble of cost-oriented, incremental and decremental SVMs is a way to address our requisites
- The promising results show the potential of optical satellite data for oil spill detection.
- In future works we will try to address the need for interpretability of the classifier, preserving in a single framework the cost-oriented and online features achieved so far.

Thanks!
Any questions?



Backup

Batch classification approach: second attempt (2/4)

Performance evaluation: ROC analysis

Performance evaluation based on classification accuracy can be inappropriate because it assumes equal error costs for misclassification of target class elements.

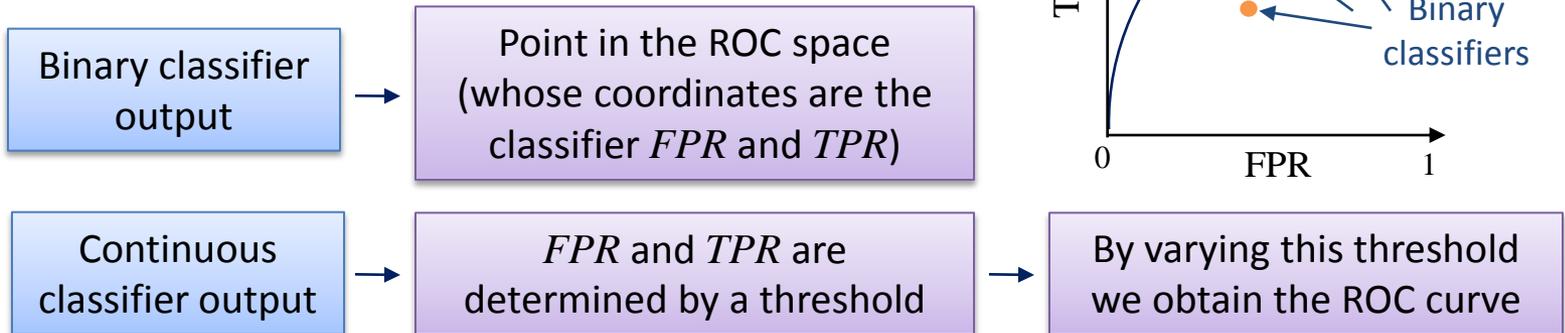
for many applications, such as oil spill detection this condition is not satisfied

In a two class classification problem, Receiver Operating Characteristic (ROC) analysis allows to graphically represent the trade-off between true positive rate and false positive rate.

The ROC space is a two dimensional space:

$$TPR = (\text{positives correctly classified}) / (\text{total positives})$$

$$FPR = (\text{negatives incorrectly classified}) / (\text{total negatives})$$



Batch classification approach: second attempt (3/4)

Performance evaluation: ROC analysis

ANFIS is a continuous classifier:

$$\Gamma(x): \mathfrak{R}^F \rightarrow [0,1].$$


 number of features

To obtain a binary classifier for the oil spill two class classification problem, we must choose a threshold τ on the classifier output.



This resulting binary classifier can be coupled with a pair (FPR, TPR)

The continuous classifier (ANFIS) can be represented by a ROC curve: each point belonging to the curve corresponds to a certain threshold, and consequently to a certain binary classifier.

The discrimination threshold can thus be chosen by observing the classifier ROC curve, basing on a trade-off between the TPR and the FPR : the target is to obtain the highest number of true positives, keeping the number of false positives as low as possible

The ROC curve is a two dimensional measure of the classifier performance. In order to have a scalar measure of it, we can consider the area under the ROC curve (AUC).

SwapOne algorithm

The ascending diagonal in the ROC space actually connects two classifiers:

- the classifier always predicting negative (the point in (0,0))
- the classifier always predicting positive (the point in (1,1))

This allows to generalize the prediction inversion procedure to linear segments connecting arbitrary classifiers \Rightarrow [algorithm SwapOne](#)

Classifier 4, built by the algorithm, has:

$$TPR_4 = TPR_1 + TPR_2 - TPR_3$$

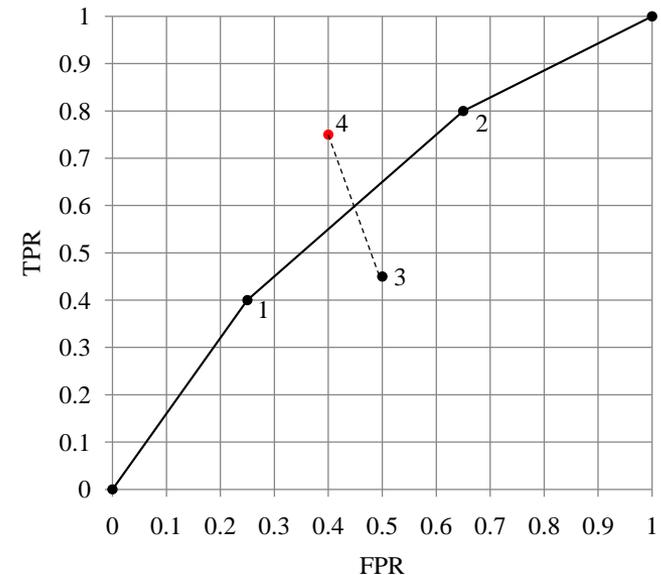
$$FPR_4 = FPR_1 + FPR_2 - FPR_3$$

This can be done if:

$$TPR_1 \leq TPR_3 \leq TPR_2$$

$$FPR_1 \leq FPR_3 \leq FPR_2$$

If classifiers 1, 2, 3 are obtained by setting different thresholds $\tau_1 > \tau_3 > \tau_2$ on the same continuous classifier, these conditions are automatically satisfied and the algorithm can be applied.



Support Vector Machines (1/3)

Cost-Oriented SVMs (CO-SVMs)

Linear separable case (simple case):

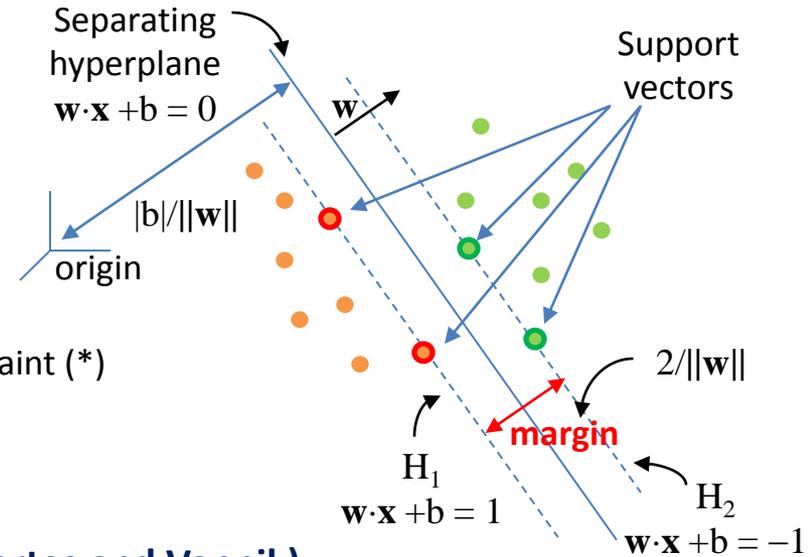
Training data: $\{x_i, y_i\}, x_i \in \mathcal{R}^F, y_i \in \{-1, 1\}$

$$\text{Constraints: } \begin{cases} x_i \cdot w + b \geq +1 & \text{for } y_i = +1 \\ x_i \cdot w + b \leq -1 & \text{for } y_i = -1 \end{cases}$$

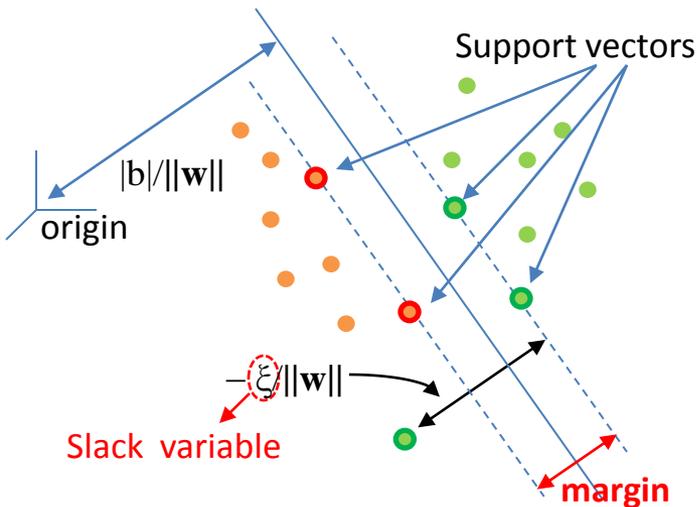
Equivalent to: $y_i (x_i \cdot w + b) - 1 \geq 0 \quad \forall i \quad (*)$

Objective function to be minimized: $\|w\|^2$ subject to constraint (*)

The SV algorithm looks for the separating hyperplane with largest margin.



Non separable case: Cost-Oriented formulation (Cortes and Vapnik)



Some errors are tolerated: some positive class elements are allowed to be on the negative side and vice versa.

$$\text{Relaxed constraints: } \begin{cases} x_i \cdot w + b \geq +1 - \xi_i & \text{for } y_i = +1 \\ x_i \cdot w + b \leq -1 + \xi_i & \text{for } y_i = -1 \end{cases} \quad \xi_i \geq 0 \quad \forall i$$

Each error has an associated cost, an increase in the objective function (proportional to ξ).
The upper bound on the maximum distance between an element and the correct hyperplane determines the cost associated with the misclassification of that pattern.

Support Vector Machines (2/3)

Non linear problems:

Non linear problems are solved by using kernel functions which map data into a Hilbert space where the problem is linear.

Incremental/Decremental SVMs (ID-SVMs) (Cauwenberghs and Poggio)

- The incremental learning algorithm builds the solution recursively by adding one new point (pattern) at a time.
- The constraints for the SVM problem are retained on the previously considered patterns, while the new point is added adiabatically to the solution.
- The decremental unlearning algorithm, in an analogous way, allows to remove data from the full trained solution.

The so built classifier is able to update the solution adapting the classification to time varying conditions.

Note: data acquisition is a batch process, while pattern evaluation is performed incrementally, following the same order as the acquisition. SVM structures are fixed by the dimension of the data acquired in batch mode

Support Vector Machines

Cost-Oriented SVMs (CO-SVMs)

Linear separable case (simple case):

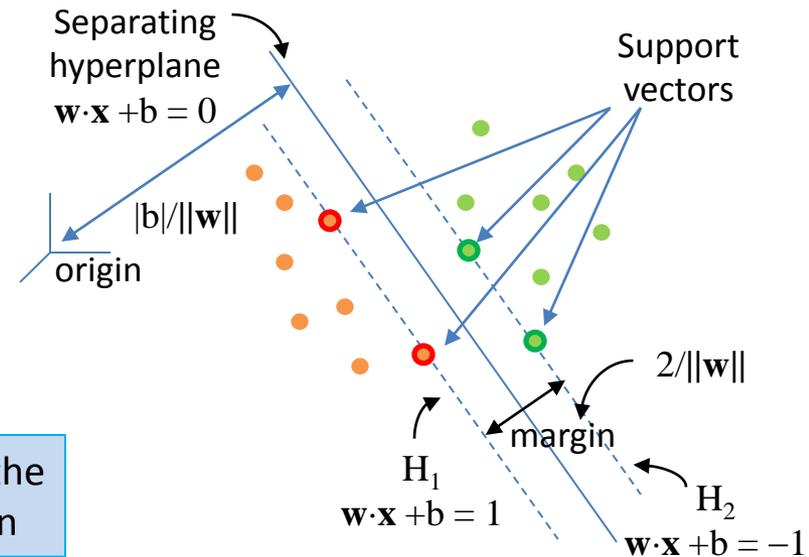
Training data: $\{\mathbf{x}_i, y_i\}$, $\mathbf{x}_i \in \mathbb{R}^F$, $y_i \in \{-1, 1\}$

Constraints: $\mathbf{x}_i \cdot \mathbf{w} + b \geq +1$ for $y_i = +1$

$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$ for $y_i = -1$

Equivalent to: $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i$

The support vector algorithm looks for the separating hyperplane with largest margin



Lagrange formulation:

$$L_P \equiv \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^l \alpha_i$$

We must now minimize L_P with respect to \mathbf{w} , b , and simultaneously require that the derivatives of L_P with respect to all the α_i vanish, with the constraint: $\alpha_i \geq 0$