Predicting the presence of oil slicks after an oil spill.

Juan Manuel Corchado and Aitor Mata.

Department of Computing Science and Automatic. University of Salamanca.
Plaza de la Merced, s/n. Salamanca. Spain.
corchado@usal.es, aitor@usal.es

Abstract. A new predicting system is presented in which the aim is to forecast the presence or not of oil slicks in a certain area of the open sea after an oil spill. In this case, the CBR methodology has been chosen to solve the problem. The system designed to predict the presence of oil slicks wraps other artificial intelligence techniques such as a Growing Radial Basis Function Networks, Growing Cell Structures and Fast Iterative Kernel Principal Components Analysis in order to develop the different phases of the CBR cycle. The proposed system uses information such as sea salinity, sea temperature, wind, currents, pressure, number and area of the slicks…. obtained from various satellites. The system has been trained using data obtained after the Prestige accident. Oil Spill CBR system (OSCBR) has been able to accurately predict the presence of oil slicks in the north west of the Galician coast, using historical data.

Keywords: Oil spill, Growing Cell Structures, Radial Basis Function, PCA.

1 Introduction

Predicting the behaviour of oceanic elements is a quite difficult task. In this case the prediction is related with external elements (oil slicks), and this makes the prediction even more difficult. Open ocean is a highly complex system that may be modelled by measuring different variables and structuring them together. Some of those variables are essential to predict the behaviour of oil slicks. In order to predict the future presence of oil slicks in an area, it is obviously necessary to know their previous positions. That knowledge is provided by the analysis of satellite images, obtaining the precise position of the slicks.

The solution proposed in this paper generates, for different geographical areas, a probability (between 0 and 1) of finding oil slicks after an oil spill. The proposed system has been constructed using historical data and checked using the data acquired during the Prestige oil spill, from November 2002 to April 2003. Most of the data used to develop the proposed system has been acquired from the ECCO (Estimating the Circulation *and Climate of the Ocean*) consortium [1]. Position and size of the slicks has been obtained by treating SAR (*Synthetic Aperture Radar*) satellite images[2].

The proposed system is a forecasting Case-Based Reasoning system: the Oil Spill CBR (*OSCBR*). A CBR system has the ability to learn from past situations, and to generate solutions to new problems based in the past solutions given to past problems. Past solutions are stored in the system, in the *case base*. In OSCBR the cases contain information about the oil slicks (size and number) as long as atmospheric data (wind, current, salinity, temperature, ocean height and pressure). OSCBR combines the efficiency of the CBR systems with artificial intelligence techniques in order to improve the results and to better generalize from past data.

The results obtained with OSCBR approximate to the real process occurred in near the ninety per cent of the value of the main variables analyzed, which is a quite important approximation.

In this paper, the oil spill problem is first presented, showing its difficulties and the possibilities of finding solutions to the problem. Then, OSCBR is explained, giving special attention to the techniques applied in the different phases of the CBR cycle. Last, the results are shown and also the future developments that can be achieved with the system.

2 Oil Spill Problem

After an oil spill, it is necessary to determine if an area is going to be contaminated or not. To conclude about the presence or not of contamination in an area it is necessary to know how the slicks generated by the spill behave. The most data available; the best solution can be given.

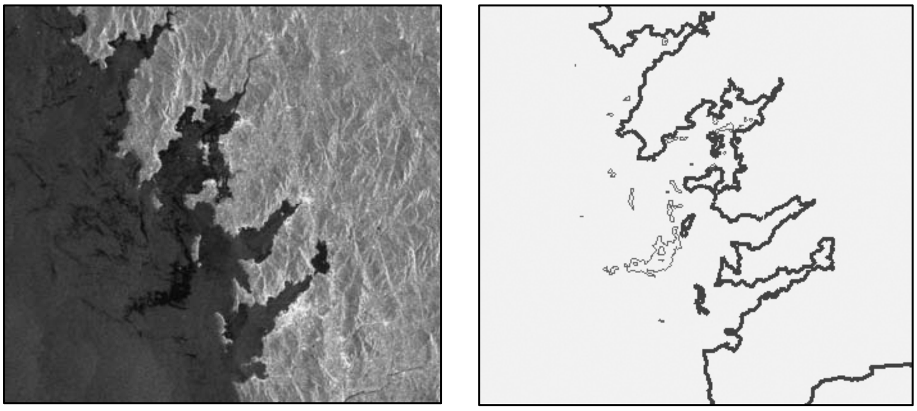


Fig. 1. On the left side, a SAR image is shown. On the right side the interpretation of the left satellite image done by OSCBR.

First, position, shape and size of the oil slicks must be identified. The most precise way to acquire that information is by using satellite images. SAR images are the most commonly used to automatically detect this kind of slicks [3]. The satellite images show certain areas where it seems to be nothing, like zone with no waves; that are the oil slicks. In *figure 1* a SAR image is shown on the left side. There, a portion of the western Galician coast is shown, as long as some black areas, corresponding to the oil slicks. The image on the right side of *figure 1* shows the interpretation of the previous image after treating the data generated by the SAR image. With SAR images it is possible to distinguish between normal sea variability and slicks. It is also important to make a distinction between oil slicks and look-alikes. Oil slicks are quite similar to quiet sea areas, so it is not always easy to discriminate between them. If there is not enough wind, the difference between the calmed sea and the surface of a slick is less evident and so, there may be more mistakes when trying to differentiate between an oil slick and something that it is not a slick. This is a crucial aspect in this problem that can also be automatically done by a series of computational tools.

Once the slicks are identified, it is also crucial to know the atmospheric and maritime situation that is affecting the slick in the moment that is being analysed. Information collected from satellites is used to obtain the atmospheric data needed. That is how different variables such as temperature, sea height and salinity are measured in order to obtain a global model [4] that can explain how slicks evolve.

2.1. Previous solutions given to the oil spill problem

There have been different ways to analyze, evaluate and predict situations after an oil spill. One approach is the simulation [5], where a model of a certain area is created, introducing specific parameters (weather, currents and wind) and working along with a forecasting system. Using this methodology, it is easy to obtain a good solution for a certain area, but it is quite difficult to generalize in order to solve the same problem in new zones.

Another way to obtain a trajectory model is to replace the oil spill by drifters [6] comparing the trajectory followed by the drifters with the already known oil slicks trajectories. If the drifters follow a similar trajectory as the one that followed the slicks, then a model can be created and there will be a possibility of creating more models in different areas. Another way of predicting oil slicks trajectories is to study previous cases to obtain a trajectory model for a certain area with different weather situations [7]. Another trajectory model is the created to accomplish the NOAA standards [8], where both the ‘best guess’ and the ‘minimum regret’ solutions are generated.

2.2. Models

One step over those solutions previously explained are the systems that, combining a major set of elements, generate response models to solve the oil spill problem.

A different of view is given by complex systems [9] that analyze large data bases (environmental, ecological, geographical and engineering), using expert systems. This way, an implicit relation between problem and solution is obtained, but with no direct connection between past examples and current decisions. Nevertheless arriving at these kind of solutions requires a great deal of data mining effort.

Once the oil spill is produced there should be contingency models that make a fast solution possible [10]. Expert systems has also been used, using the stored information from past cases, as a repository where future applications will find structured information. Some other complete models have been created, to integrate the different variables affecting the spills [11], always trying to get better benefits than the possible costs generated by all the infrastructure needed to response to a problematic generated situation

The final objective of all these systems is to be decision support systems, in order to help to take all the decisions that need to be taken properly organized. To achieve that great objective, different techniques have been used, from fuzzy logic [12] to negotiation with multi-agent systems [13]

3 Oil Spill CBR system – OSCBR

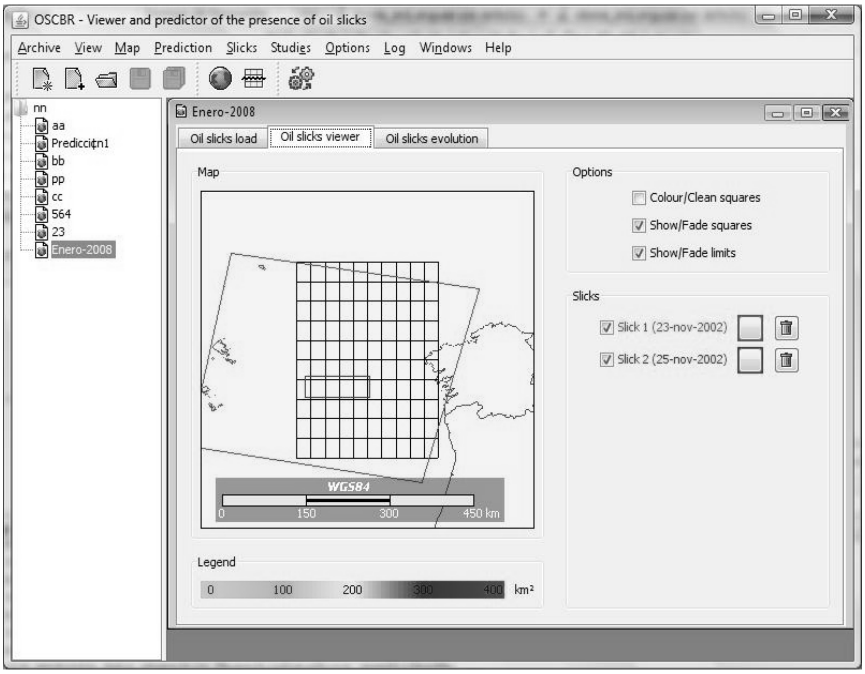
Case-Based reasoning is a methodology [14], and so it has been applied to solve different kind of problems, from health applications [15; 16] to eLearning [17] [18]. CBR has also evolved, being transformed so that it can be used to solve new problems, becoming a methodology to plan [19], or distributed version [20].

Table 1. Variables that define a case .

Variable	Definition	Unit
Longitude	Geographical longitude	Degree
Latitude	Geographical latitude	Degree
Date	Day, month and year of the analysis	dd/mm/yyyy
Sea Height	Height of the waves in open sea	m
Bottom pressure	Atmospheric pressure in the open sea	Newton/m^2
Salinity	Sea salinity	ppt (parts per thousand)
Temperature	Celsius temperature in the area	ºC
Area of the slicks	Surface covered by the slicks present in the analyzed area	Km^2
MeridionalWind	Meridional direction of the wind	m/s
Zonal Wind	Zonal direction of the wind	m/s
Wind Strenght	Wind strength	m/s
Meridional Current	Meridional direction of the ocean current	m/s
Zonal Current	Zonal direction of the ocean current	m/s
Current Strenght	Ocean current strength	m/s

CBR has already been used to solve maritime problems [21] in which different oceanic variables were involved. In this case, the data collected from different observations from satellites, is pre-processed, and structured in cases. The created cases are the keys to obtain the solutions to future problems, through the CBR system. Oil slicks are detected using SAR images. Those images are processed and transformed to be used by the system.

Figure 2 shows the graphical user interface of the developed system. In that image the different components of the application can be seen (maps, prediction, slicks, studies…) as well as a visualization of an oceanic area with oil slicks and a squared area to be analyzed.



☐ Colour/Clean squares

☑ Show/Fade squares

☑ Show/Fade limits

☑ Slick 1 (23-nov-2002)

☑ Slick 2 (25-nov-2002)

Fig. 2. Graphical user interface of the OSCBR system. The different components of the system can be observed here.

OSCBR determines the probability of finding oil slicks in a certain area. To generate the predictions, the system divides the area to be analyzed in squares of approximately half a degree side. Then the system determines the amount of slicks present in a square. The squares where the slicks are located are coloured with different gradation depending on the quantity of the squared area covered by oil slicks.

The squared zone determines the area that is going to be analyzed independently. The values of the different variables in a square area in a certain moment as long as the value of the possibility of finding oil slicks in the following day is what is called *a case*, which define the problem and propose the solution.

In *table 1* the structure of a case is shown. The variables present in a case can be geographical (longitude and latitude), temporal (date of the case), atmospheric (wind, current, sea height, bottom pressure, salinity and temperature) and variables directly related with the problem (number and area of the slicks).

Once the data is structured, it is stored in the *case base*. Every case has its temporal situation stored and that relates every case with the next situation in the same position. That temporal relationship is what creates the union between *problem* and *solution*. The problem is the past case, and the solution is the future case, the future state of the square analyzed.

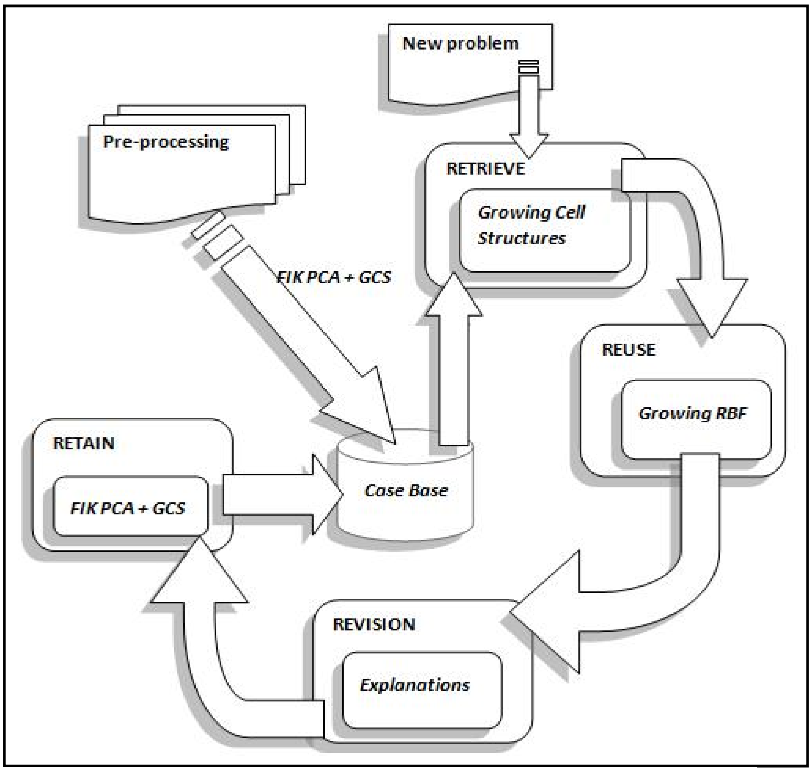


Fig 3. CBR cycle adapted to the OSCBR system.

The data used to train the system has been obtained after the Prestige accident, between November 2002 and April 2003, in a specific geographical area to the north west of the Galician coast (longitude between 14 and 6 degrees west and latitude between 42 and 46 degrees north). When all that information is stored in the case base, the system is ready to predict future situations. To generate a prediction, a problem situation must be introduced in the system. Then the most similar cases to the problematic situation are retrieved from the case base. Once a collection of cases are chosen from the case base, they must be used to generate the solution to the current problem. *Growing Radial Basis Functions Networks* [22] are used to combine the chosen cases in order to obtain the new solution.

OSCBR uses both the capabilities of a standard CBR system and the power of artificial intelligence techniques. As shown in *figure 3*, every CBR phase uses an artificial intelligence technique in order to obtain its solution. In *figure 3* the four main phases of the CBR cycle are shown as long as the AI techniques used in each phase. Those phases with its related techniques are explained next.

3.1 Pre-processing

Historical data collected from November 2002 to april 2003 is used to create the *case base*. As explained before, cases are formed by a series of variables. *Principal Components Analysis* (PCA) [23] can reduce the number of those variables and then, the system stores the value of the principal components, which are related with the original variables that define a case. PCA has been previously used to analyse oceanographic data and it has proved to be a consistent technique when trying to reduce the number of variables.

In this paper *Fast Iterative Kernel PCA*(FIKPCA), an evolution of PCA, has been used [24]. This technique reduces the number of variables in a set by eliminating those that are linearly dependent, and it is quite faster than the traditional PCA. To improve the convergence of the Kernel Hebbian Algorithm used by Kernel PCA, FIK-PCA set η_t proportional to the reciprocal of the estimated eigenvalues. Let $\lambda_t \in \Re^r_+$ denote the vector of eigenvalues associated with the current estimate of the first *r* eigenvectors. The new KHA algorithm sets de i^{th} component of η_t to the files.

$$[\eta_t]_i = \frac{1}{[\lambda_t]_i}\frac{\tau}{t+\tau}\eta_0 \ , \tag{1}$$

The final variables are, obviously, linearly independent and are formed by combination of the previous variables. The values of the original variables can be recovered by doing the inverse calculation to the one produced to obtain the new variables. The variables that are less used in the final stored variables are those whose values suffer less changes during the periods of time analysed (salinity, temperature and pressure do not change from one day to another, then, they can be ignored considering that the final result does not depend on them).

Once applied the FIKPCA, the number of variables is reduced to three, having the following distribution:

Variable_1: -0,560 * long - 0,923*lat + 0,991*s_height + 0,919*b_pressure + 0,992*salinity + 0,990*temp - 0,125*area_of_slicks + 0,80*mer_wind + 0,79*zonal_wind + 0,123*w_strenght + 0,980*mer_current + 0,980*zonal_current + 0,980*c_strength

Variable_2: 0,292*long - 0,081*lat - 0,010*s_height - 0,099*b_pressure - 0,011*salinity - 0,013*temp - 0,021*area_of_slicks + 0,993*merl_wind + 0,993*zonal_wind + 0,989*w_strenght - 0,024*mer_current - 0,024*zonal_current - 0,024*c_strength

Variable_3: 0*long - 0,072*lat + 0,009*s_height + 0,009*b_pressure + 0,009*salinity + 0,009*temp + 0,992*area_of_slicks + 0,006*mer_wind + 0,005*zonal_wind + 0,005*w_strenght - 0,007*mer_current - 0,007*zonal_current - 0,007*c_strength

After applying FIKPCA, the historical data is stored in the case base, and is used to solve future problems using the rest of the CBR cycle. Storing the principal components instead of the original variables implies reducing the amount of memory necessary to store the information in about a sixty per cent which is more important as the case base grows. The reduction of the number of variables considered also implies a faster recovery from the case base.

When introducing the data into the case base, *Growing Cell Structures* [25] are used. GCS can create a model from a situation organizing the different cases by their similarity. If a 2D representation is chosen to explain this technique, the most similar cells (*cases* in OSCBR) are near one of the other. If there is a relationship between the cells, they are grouped together, and this grouping characteristic helps the CBR system to recover the similar cases in the next phase. When a new cell is introduced in the structure, the closest cells move towards the new one, changing the overall structure of the system. The weights of the winning cell, ω_c, and its neighbours, ω_n, are changed. The terms ε_c and ε_n represent the learning rates for the winner and its neighbours, respectively. x represents the value of the input vector.

$$\omega_c(t+1) = \omega_c(t) + \varepsilon_c(x - \omega_c) \tag{2}$$

$$\omega_n(t+1) = \omega_n(t) + \varepsilon_n(x - \omega_n) \tag{3}$$

The pseudocode of the GCS insertion process is shown below:

```
1. The most similar cell to the new one is found.
2. The new cell is introduced in the middle of the connection
   between the most similar cell and the least similar to the
   new one.
3. Direct neighbours of the closest cell change their values by
   approximating to the new cell and specified percentage of
   the distance between them and the new cell.
```

3.2 Retrieve

Once the case base has stored the historical data, and the GCS has learned from the original distribution of the variables, the system is ready to receive a new problem.

When a new problem comes to the system, GCS are used once again. The stored GCS behaves as if the new problem would be stored in the structure, and finds the most similar cells (cases in the CBR system) to the problem introduced in the system. In this case the GCS does not change its structure, because it is being used to obtain the most similar cases to the introduced problem. Only in the retain phase, the GCS changes again, introducing if it is correct, the proposed solution.

The similarity of the new problem to the stored cases is determined by the GCS calculating the distance between them. Every element in the GCS has a series of values (every value corresponds to one of the principal components created after de PCA analysis) and then the distance between elements is a multi-dimensional distance, where all the elements are considered to establish the distance between cells.

Then, after obtaining the most similar cases from the case base, they are used in the next phase. The most similar cases stored in the case base will be used to obtain an accurate prediction according to the previous solutions related with the selected cases.

3.3 Reuse

Once the most similar cases to the problem to be solved are recovered from the case base, they are used to generate the solution. The prediction of the future probability of finding oil slicks in an area is generated using an artificial neural network, with a hybrid learning system. An adaptation of Radial Basis Functions Networks are used to obtain that prediction [26]. The chosen cases are used to train the artificial neural network. Radial Basis Function networks have been chosen because of the reduction of the training time comparing with other artificial neural network systems, such as Multilayer Perceptrons. In this case, in every analysis the network is trained, using only the cases selected from the case base, the most similar to the proposed problem.

Growing RBF networks [27] are used to obtain the predicted future values corresponding to the proposed problem. This adaptation of the RBF networks allows the system to grow during training gradually increasing the number of elements (prototypes) which play the role of the centers of the radial basis functions. In this case the creation of the Growing RBF must be made automatically, which implies an adaptation of the original GRBF system. The pseudocode of the growing process and the definition of the error for every pattern is shown below:

$$e_i = {}^{l}\!/_{p^*} \sum_{k=1}^{p} ||t_{ik} - y_{ik}|| \, , \tag{4}$$

Where t_{ik} is the desired value of the k^{th} output unit of the i^{th} training pattern, y_{ik} the actual values ot the k^{th} output unit of the i^{th} training pattern.

Growing RBF pseudocode:

```
1. Calculate the error, $e_i$ (4) for every new possible prototype.
      a. If the new candidate does not belong to the chosen
         ones and the error calculated is less than a threshold
         error, then the new candidate is added to the set of
         accepted prototypes.
      b. If the new candidate belongs to the accepted ones and
         the error is less than the threshold error, then
         modify the weights of the neurons in order to adapt
         them to the new situation.
2. Select the best prototypes from the candidates
      a. If there are valid candidates, create a new cell
         centered on it.
      b. Else, increase the iteration factor. If the iteration
         factor comes to the 10% of the training population,
         freeze the process.
3. Calculate global error and update the weights.
      a. If the results are satisfactory, end the process. If
         not, go back to step 1.
```

Once the GRBF network is created, it is used to generate the solution to the proposed problem. The solution will be the output of the network using as input data the selected cases from the case base.

3.4 Revise

After generating the prediction, it is shown to the user in a similar way the slicks are interpreted by OSCBR. A set of squared coloured areas appear. The intensity of the colour corresponds with the possibility of finding oil slicks in that area. The areas coloured with a higher intensity are those with the highest probability of finding oil slicks in them.

In this visual approximation, the user can check if the solution is a good one or not. The system also provides an automatic method of revision that must be, anyway, checked by an expert user, confirming the automatic revision.

Explanations are used to check the correction of the proposed solution, to justify the solution. To obtain a justification to the given solution, the cases selected from the case base are used once again. To create an *explanation,* a comparison between different possibilities has been used [28].All the selected cases has its own *future situation* associated. If we consider the case and its solution as two vectors, we can establish a *distance* between them, calculating the evolution of the situation in the considered conditions. If the distance between the proposed problem and the solution given is not greater than the distances obtained from the selected cases, then the solution is a good one, according to the structure of the case base.

Explanation pseudocode:

```
1.For every selected case in the retrieval phase, the distance
  between the case and its solution is calculated.
2.The distance between the proposed problem and the proposed
  solution is also calculated.
3.If the difference between the distance of the proposed
  solution and those of the selected cases is below a certain
  threshold value, then the solution is considered as a valid
  one.
4.If not, the user is informed and the process goes back to the
  retrieval phase, where new cases are selected from the case
  base.
5.If, after a series of iterations the system does not produce a
  good enough solution, then the user is asked to consider the
  acceptance of the best of the generated solutions.
```

The distances are calculated considering the sign of the values, not using its absolute value. This decision is easily justified by the fact that is not the same to move to the north than to the south, even if the distance between two points is the same. If the prediction is considered as correct it will be stored in the case base, and it can then be used in next predictions to obtain new solutions.

3.5 Retain

When the proposed prediction is accepted, it is considered as a good solution to the problem and can be stored in the case base in order to serve to solve new problems. It will have the same category as the historical data previously stored in the system.

When inserting a new case in the case base, *Fast Iterative Kernel PCA* is once used to reduce the number of variables used and to adapt the data generated by the system. The adaptation is done by changing the original variables into the principal components previously chosen by the system.

Obviously, when introducing a new case in the case base, the GCS formed by the information stored in the case base, also change, to adapt to the new situation generated. When adapting to the new solution introduced in the case base, the GCS system grows and improves its capability of generating good results as new knowledge is introduced in the system.

4 Results

The historical data used to train the system has been obtained from different satellites. Temperature, salinity, bottom pressure, sea height, wind, currents, number and area of the slicks, as long as the location of the squared area and the date have been used to create a case. All these data define the problem case and also the solution case. The solution to a problem defined by an area and its variables is the same area, but with the values of the variables changed to the prediction obtained from the CBR system.

When the OSCBR system has been used with a subset of the data that has not been previously used to train the system, it has produced encouraging results. The predicted situation was contrasted with the actual future situation. The future situation was known, as long as historical data was used to develop the system and also to test the correction of it. The proposed solution was, in most of the variables, close to 90% of accuracy.

Table 2. Percentage of good predictions obtained with different techniques.

Number of cases	RBF	CBR	RBF + CBR	OSCBR
100	45 %	39 %	42 %	43 %
500	48 %	43 %	46 %	46 %
1000	51 %	47 %	58 %	64 %
2000	56 %	55 %	65 %	72 %
3000	59 %	58 %	68 %	81 %
4000	60 %	63 %	69 %	84 %
5000	63 %	64 %	72 %	87 %

For every problem, defined by an area and its variables, the system offers nine solutions: the same area, with its proposed variables and the eight closest neighbours. This way of prediction is used in order to clearly observe the direction of the slicks, what can be useful in order to determine the coastal areas that will be affected by the slicks generated after an oil spill.

In *table 2* a summary of the results obtained is shown. In this table different techniques are compared. The table shows the evolution of the results along with the increase of the number of cases stored in the case base. All the techniques analyzed improve its results when increasing the number of cases stored. Having more cases in the case base, makes easier to find similar cases to the proposed problem and then, the solution can be more accurate. The "*RBF"* column represents a simple Radial Basis Function Network that is trained with all the data available. The network gives an output that is considered a solution to the problem. The "*CBR"* column represents a pure CBR system, with no other techniques included, the cases are stored in the case bases and recovered considering the Euclidean distance. The most similar cases are selected and after applying a weighted mean depending on the similarity, a solution s proposed. The *"RBF + CBR"* column corresponds to the possibility of using a RBF system combined with CBR. The recovery from the CBR is done by the Manhattan distance and the RBF network works in the reuse phase, adapting the selected cases to obtain the new solution. The results of the "*RBF+CBR"* column are, normally, better than those of the "*CBR"*, mainly because of the elimination of useless data to generate the solution. Finally, the "*OSCBR"* column shows the results obtained by the proposed system , obtaining better results that the three previous analyzed solutions.

Table 3 shows a multiple comparison procedure (*Mann-Whitney* test) used to determine which models are significantly different from the others.

The asterisk indicates that these pairs show statistically significant differences at the 99.0% confidence level. It can be seen in *table 3*, that the OSCBR system presents statistically significant differences with the rest of the models.

The proposed solution does not generate a trajectory, but a series of probabilities in different areas, what is far more similar to the real behaviour of the oil slicks.

Once the prediction is generated and approved, it can be exported to various formats. First an html file can be generated with the images that represent the prediction, the solution to the problem. Other output formats are "*Google related":* the solutions can be exported to *Google Earth* and to *Google Maps.*

Table 3. Multiple comparison procedure among different techniques.

	RBF	CBR	RBF + CBR	OSCBR
RBF				
CBR	*			
RBF+CBR	=	=		
OSCBR	*	*	*	

5 Conclusions and future work

In this paper, the OSCBR system has been explained. It is a new solution for predicting the presence or not of oil slicks in a certain area after an oil spill.

This system used data acquired from different orbital satellites and with that data the CBR environment was created. The data must be previously classified into the structure required by the CBR system to store it as a case.

OSCBR uses different artificial intelligence techniques in order to obtain a correct prediction. *Fast Iterative Kernel Principal Component Analysis* is used to reduce the number of variables stored in the system, getting about a 60% of reduction in the size of the *case base.* This adaptation of the PCA also implies a faster recovery of cases from the case base (more than 7% faster than storing the original variables).

To obtain a prediction using the cases recovered from the case base, *Growing Radial Basis Function Networks* has been used. This evolution of the RBF networks implies a better adaptation to the structure of the case base, which is organised using *Growing Cell Structures.* The results using Growing RBF networks instead of simple RBF networks are about a 4% more accurate, which is a good improvement.

Evaluations show that the system can predict in the conditions already known, showing better results than previously used techniques. The use of a combination of techniques integrated in the CBR structure makes it possible to obtain better result than using the CBR alone (17% better), and also better than using the techniques isolated, without the integration feature produced by the CBR (11% better).

The next step is generalising the learning, acquiring new data to create a base of cases big enough to have solutions for every season. Another improvement is to create an on-line system that can store the case base in a server and generate the solutions dynamically to different requests. This on-line version will include real time connection to data servers providing weather information of the current situations in order to predict real future situations.

References

1. Menemenlis, D., Hill, C., Adcroft, A., Campin, J.M., et al. (2005) NASA Supercomputer Improves Prospects for Ocean Climate Research, EOS Transactions, 86 (9), 89-95.
2. Palenzuela, J.M.T., Vilas, L.G. and Cuadrado, M.S. (2006) Use of ASAR images to study the evolution of the Prestige oil spill off the Galician coast, International Journal of Remote Sensing, 27 (10), 1931-1950.
3. Solberg, A.H.S., Storvik, G., Solberg, R. and Volden, E. (1999) Automatic detection of oil spills in ERS SAR images, IEEE Transactions on Geoscience and Remote Sensing, 37 (4), 1916-1924.
4. Stammer, D., Wunsch, C., Giering, R., Eckert, C., et al. (2003) Volume, heat, and freshwater transports of the global ocean circulation 1993–2000, estimated from a general circulation model constrained by World Ocean Circulation Experiment (WOCE) data, Journal of Geophysical Research, 108 (10.1029).
5. Brovchenko, I., Kuschan, A., Maderich, V. and Zheleznyak, M. (2002) The modelling system for simulation of the oil spills in the Black Sea, 3rd EuroGOOS Conference: Building the European capacity in operational oceanography., 192.
6. Price, J.M., Ji, Z.G., Reed, M., Marshall, C.F., et al. (2003) Evaluation of an oil spill trajectory model using satellite-tracked, oil-spill-simulating drifters, OCEANS 2003. Proceedings, 3.
7. Vethamony, P., Sudheesh, K., Babu, M.T., Jayakumar, S., et al. (2007) Trajectory of an oil spill off Goa, eastern Arabian Sea: Field observations and simulations, Environmental Pollution.

8. Beegle-Krause, C.J. (1999) GNOME: NOAA's next-generation spill trajectory model, OCEANS'99 MTS/IEEE. Riding the Crest into the 21st Century, 3, 1262-1266.
9. Douligeris, C., Collins, J., Iakovou, E., Sun, P., et al. (1995) Development of OSIMS: An oil spill information management system, Spill Science & Technology Bulletin, 2 (4), 255-263.
10. Reed, M., Ekrol, N., Rye, H. and Turner, L. (1999) Oil Spill Contingency and Response (OSCAR) Analysis in Support of Environmental Impact Assessment Offshore Namibia, Spill Science and Technology Bulletin, 5 (1), 29-38.
11. Belore, R. (2005) The SL Ross oil spill fate and behavior model: SLROSM". Spill Science and Technology Bulletin.
12. Liu, X. and Wirtz, K.W. (2007) Decision making of oil spill contingency options with fuzzy comprehensive evaluation, Water Resources Management, 21 (4), 663-676.
13. Liu, X. and Wirtz, K.W. (2005) Sequential negotiation in multiagent systems for oil spill response decision-making, Marine Pollution Bulletin, 50 (4), 469-474.
14. Watson, I. (1999) Case-based reasoning is a methodology not a technology, Knowledge-Based Systems, 12 (5-6), 303-308.
15. Montani, S., Portinale, L., Leonardi, G. and Bellazzi, R. (2006) Case-based retrieval to support the treatment of end stage renal failure patients, Artificial Intelligence in Medicine, 37 (1), 31-42.
16. Corchado, J.M., Bajo, J. and Abraham, A. (2008) GERAmI: Improving the delivery of health care., IEEE Intelligent Systems. Special Issue on Ambient Intelligence.
17. Decker, B., Rech, J., Althoff, K.D., Klotz, A., et al. (2005) eParticipative Process Learning—process-oriented experience management and conflict solving, Data & Knowledge Engineering, 52 (1), 5-31.
18. Althoff, K.D., Mänz, J. and Nick, M. (2005) Maintaining Experience to Learn: Case Studies on Case-Based Reasoning and Experience Factory. Proc. 6th Workshop Days of the German Computer Science Society (GI) on Learning, Knowledge, and Adaptivity (LWA 2005) Saarland University, Germany.
19. Cox, M.T., MuÑOz-Avila, H. and Bergmann, R. (2006) Case-based planning, The Knowledge Engineering Review, 20 (03), 283-287.
20. Plaza, E. and McGinty, L. (2006) Distributed case-based reasoning, The Knowledge Engineering Review, 20 (03), 261-265.
21. Corchado, J.M. and Fdez-Riverola, F. (2004) FSfRT: Forecasting System for Red Tides, Applied Intelligence, 21, 251-264.
22. Karayiannis, N.B. and Mi, G.W. (1997) Growing radial basis neural networks: merging supervised andunsupervised learning with network growth techniques, Neural Networks, IEEE Transactions on, 8 (6), 1492-1506.
23. Dunteman, G.H. (1989) Principal Components Analysis. Newbury Park, California.
24. Gunter, S., Schraudolph, N.N. and Vishwanathan, S.V.N. (2007) Fast Iterative Kernel Principal Component Analysis, Journal of Machine Learning Research, 8, 1893-1918.
25. Fritzke, B. (1994) Growing cell structures—a self-organizing network for unsupervised and supervised learning, Neural Networks, 7 (9), 1441-1460.

26. Haykin, S. (1999) Neural networks. Prentice Hall Upper Saddle River, NJ.
27. Ros, F., Pintore, M. and Chrétien, J.R. (2007) Automatic design of growing radial basis function neural networks based on neighboorhood concepts, Chemometrics and Intelligent Laboratory Systems, 87 (2), 231-240.
28. Plaza, E., Armengol, E. and Ontañón, S. (2005) The Explanatory Power of Symbolic Similarity in Case-Based Reasoning, Artificial Intelligence Review, 24 (2), 145-161.