Case-Based Collective Inference for Maritime Object Classification

Kalyan Moy Gupta[1], David W. Aha[2], and Philip Moore[1]

[1]Knexus Research Corporation; Springfield, VA 22153
[2]Navy Center for Applied Research in Artificial Intelligence;
Naval Research Laboratory (Code 5514); Washington, DC 20375
firstname.lastname@knexusresearch.com | david.aha@nrl.navy.mil

Abstract. Maritime assets such as merchant and navy ships, ports, and harbors, are targets of terrorist attacks as evidenced by the USS Cole bombing. Conventional methods of securing maritime assets to prevent attacks are manually intensive and error prone. To address this shortcoming, we are developing a decision support system that shall alert security personnel to potential attacks by automatically processing maritime surveillance video. An initial task that we must address is to accurately classify maritime objects from video data, which is our focus in this paper. Object classification from video images can be problematic due to noisy outputs from image processing. We approach this problem with a novel technique that exploits maritime domain characteristics and formulates it as a graph of spatially related objects. We then apply a case-based *collective classification* algorithm on the graph to classify objects. We evaluate our approach on river traffic video data that we have processed. We found that our approach significantly increases classification accuracy in comparison with a conventional (i.e., non-relational) alternative.

1 Introduction

Maritime assets such as merchant and naval vessels are under a constant threat of terrorist attacks. For example, the USS Cole was completely disabled in a bombing event at the Port of Yemen that claimed the lives of 11 sailors.[1] Existing approaches to counter such threats use a combination of sensors such as radar, video surveillance, and manual *watchstanding*. These approaches are manually intensive; potential threats can be overlooked due to human factors such as information overload and fatigue. In addition, sensors such as radar are largely ineffective against small, fast moving vessels. We are developing a decision support system, named the *Maritime Activity Analysis Workbench* (MAAW), to address some of these problems. MAAW is being designed to detect potentially threatening surface vessels by automatically processing maritime surveillance video. A critical task in this context is that of classifying maritime objects. If accurately predicted, MAAW can then assess their potential threat and issue alerts to *watchstanders*, our primary end users.

Our object classifier must operate on potentially noisy data obtained from image processing components yet perform robustly. When combined with a large number of closely related vessel types, this poses a significant challenge for conventional

[1] http://en.wikipedia.org/wiki/USS_Cole_bombing

classification methods, which classify objects independently. However, in the maritime domain, we expect the context of maritime objects to provide important clues for object classification. For example, a tugboat in a harbor often tows a cargo vessel. Given that images of small vessels are harder to accurately classify than those of large cargo vessels, their proximity could be an important clue for classifying tugboats. One approach for incorporating contextual clues is *collective classification*. Unlike conventional approaches, collective classifiers ***concurrently*** classify a set of ***related*** objects. This approach has not previously been applied to the task of maritime object classification, and its effectiveness on this task is unknown.

Our focus in this paper is an object classification method that exploits contextual cues in a maritime video scene. Our contributions are as follows. First, we frame our task as that of using contextual relational cues to increase object classification accuracy. We represent these cues by transforming a maritime scene into a graph of spatially related objects. We then apply a case-based collective classifier, which includes a conventional classifier, domain-specific parameterized similarity measures (learned from the data), and a collective inference procedure. Finally, we evaluate our approach on maritime river traffic video captured by our system. We found that case-based collective classification significantly outperforms a conventional independent object classification approach on this task.

We briefly describe the topic of maritime surveillance and related work in Section 2. In Section 3, we present an overview of our system. In Section 4, we describe its algorithm for case-based collective classification of maritime objects. We discuss the evaluation of our approach in Section 5. Finally, we conclude the paper with remarks on future research.

2 Maritime Video Surveillance and Related Work

Other researchers have addressed maritime domain awareness tasks not unlike the port/harbor security task that is our ultimate focus. For example, Rhodes et al. (2005) employ neural network classifiers (specifically, a modification of Fuzzy ARTMAP) to learn normalcy models for anomaly detection in maritime environments. As with MAAW, their objective is for operators to provide feedback for learning. We are addressing the task of using spatial and temporal relations to detect coordinated activities. However, their data combines metadata with automated identification systems (AIS) data, rather than video imagery.

ObjectVideo[2] deploys sophisticated products for maritime (and other types of) intelligence surveillance, including those for real-time, high-speed, activity-based video indexing and retrieval. This can be used, for example, to perform forensic analysis (e.g., detect the movement of suspicious objects). Motivated by the fact that humans cannot monitor a vast number of vessels/objects simultaneously, their system automatically extracts object descriptions (from a variety of sensors) using a statistical pixel modeling approach, and employs user-provided rules to determine when to generate security alerts (Lipton *et al.*, 2009). A key difference of our approach is that we instead use a case-based statistical relational learning approach for object recognition.

[2] http://www.objectvideo.com

There is a long and rich history of research in case-based reasoning (CBR) on image processing (Perner *et al.*, 2005). Most of these have focused on images rather than video. For example, Perner's (1993) cases are graphically represented using spatial relations, and structural similarity is computed to classify weld defects. In contrast, our cases are *related* spatially and are not represented using only intrinsic attributes. Also, we instead use collective inference for object classification.

Work on video data within CBR has primarily concerned methods for video retrieval. For example, Burke and Kass (1995) describe an approach for retrieving and presenting stories from a video data base to support educational goals. Similarly, Johnson et al. (2000) describe an ASK system (the Air Campaign Planning Advisor) in which educational video recordings of domain experts can be retrieved through a tailored interface and a hierarchical task decomposition model. Zhang and Nunamaker (2004) index videos by their metadata. Their system retrieves cases using natural language queries. While this genre of research focuses on user interfaces and video retrieval, our focus in this paper is instead on object recognition in video data.

Some other uses of video have been the focus of CBR and related research. For example, MacNeil (1991) describes a visual programming environment in which CBR is used to capture and reuse knowledge pertaining to the creation of multimedia presentations. In MacNeil's TYRO, cases are generic temporal templates, abstracted from video segments, which denote chunks of design experience. These general cases can then be used to provide constraints for creating similar videos. In contrast, we focus on specific case representations, and recognizing objects from video.

Our ultimate focus is on threat analysis: if we can accurately identify the objects in these videos and recognize their behaviors, our next step will be to assess whether these behaviors are threatening (i.e., to naval/maritime assets). A variety of CBR research has focused on threat analysis techniques. For example, Jung et al. (1999) use CBR to perform risk analysis for organizations in electronic commerce environments, where cases are used to evaluate assets and assess threats and vulnerabilities by comparison with prior security accidents. Multiple groups have also used CBR to assist intelligence analysts with constructing (Adams & Goel, 2007) and/or analyzing (Murdock *et al.*, 2003) hypothesized terrorist threats. CBR has also been used, several times, to detect anomalies in computer networks (e.g., see a recent investigation described by Micarelli and Sansonetti (2007)). A primary distinction of our work from these investigations is that we are working with video data.

Finally, a distinguishing feature of our approach is that we use collective classification to leverage the spatial and temporal relations among objects to increase predictive accuracy. Previous work on collective classification, a form of statistical relational learning (SRL), has not been applied to tasks involving video data (Sen *et al.*, 2008). This includes our own previous research on case-based collective classification (McDowell *et al.*, 2007a). However, other SRL approaches have been applied to similar tasks. In particular, Aboutalib and Veloso (2007) leverage human-provided cues, detected from humans interacting with objects in video data, to recognize those objects using probabilistic relational models. Unlike our work, they do not use a CBR approach for this task, nor focus on maritime object recognition.

3 The Maritime Activity Analysis Workbench

Our goal is to develop a decision support system for maritime security personnel (e.g., watchstanders, harbor masters) to assist them with their surveillance and decision making tasks and improve their threat assessment capability. To meet this goal, we are developing MAAW. It includes a series of adaptive processors, ranging from video acquisition to threat analysis, designed to interact with its user to issue alerts, provide threat assessments, and receive performance feedback with corrections (see Figure 1). A crucial component in its pipeline of processors is the Object Classifier, which we discuss in detail in Section 4. Here, we briefly review MAAW's intended functionality and explain the role of its Object Classifier.

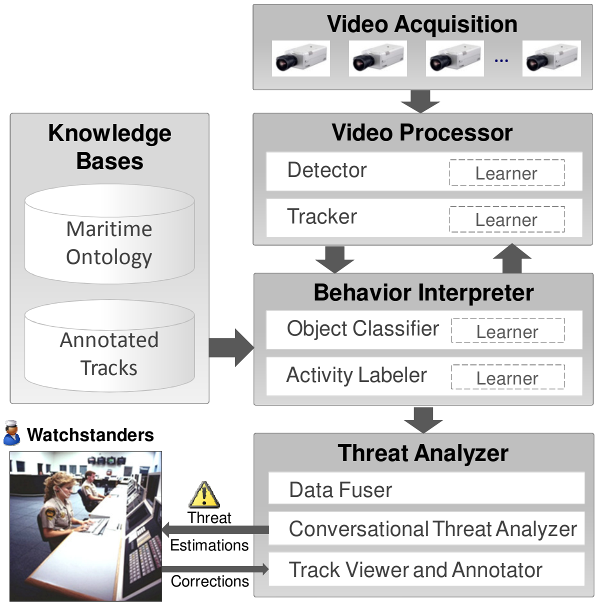


Figure 1: MAAW's Functional Architecture

MAAW's *Video Acquisition* subsystem currently includes a fixed video camera that captures maritime traffic video in black and white, digital format. It can also capture videos from online harbor cams. The acquisition system suitably compresses the video and hands it off to the *Video Processor* for further processing. The *Detector* within the Video Processor performs basic operations such as adaptive background subtraction to detect moving maritime objects such as boats and ships. The *Tracker*, also a component of the Video Processor, then groups the objects detected from a series of video frames into tracks. It uses a combination of *Appearance Models* (not shown in Figure 1) and clustering techniques to perform its task.

The Video Processor outputs track information in a data structure suitable for consumption by the *Behavior Interpreter*. The track is represented as a series of events, each referring to a maritime object and its attributes such as position, speed, and image signature. The Behavior Interpreter's function is to classify the objects within a track and the activities they are performing. We are combining supervised learning approaches along with maritime surveillance domain knowledge to accomplish these tasks. The *Object Classifier* and the *Activity Labeler* are the two components in the Behavior Interpreter. They rely on two knowledge bases: a *Maritime Ontology*, and a database of *Annotated Tracks*. We are annotating all the track data with objects categories and activity descriptions chosen from this ontology, and expect users and subject matter experts to do so minimally while using MAAW.

The maritime object classification task can be challenging because tracks extracted by the Video Processor can be noisy depending on a variety of application conditions such as the weather, time of day, the size and the number of objects, and occlusion. For example, a single object in the scene could result in multiple spurious tracks with inaccurate attribute value estimates. In this paper, we explore one way to address this problem. We investigate whether taking application and scene context into account can increase classification accuracy, even when the track data is noisy.

The Behavior Interpreter hands off the automatically labeled tracks to the *Threat Analyzer*, which will fuse the labeled tracks with harbor database information to assess threats and issue alerts. End users will be able to accept or reject MAAW's decisions and provide corrective feedback, which MAAW will use to update the track database.

MAAW is, in part, an annotation tool for video processing. Rincón and Martínez-Cantos (2007) survey other such tools, which differ from MAAW in that they do not employ case-based collective classification to perform maritime object classification.

4 Case-Based Collective Classification of Maritime Objects

4.1 Collective Classification

Conventional classification methods assume that cases/instances are independently drawn from an identical distribution (the i.i.d. assumption), and are otherwise not related. However, in many practical classification tasks cases can be explicitly or implicitly related. For example, web pages can be explicitly related by hyperlinks and emails can be implicitly related as requests and responses. Likewise, in a maritime object classification task, cases that represent objects can be implicitly related spatially and/or temporally. For example, a tugboat track can be spatially related to a cargo-ship track that it is towing out of a harbor.

Relations across cases can provide valuable contextual information that can potentially be leveraged to increase classification accuracy. For example, *collective classifiers* use these relations to concurrently classify cases, which can often increase classification accuracy (Sen *et al.,* 2008). The magnitude of accuracy increase depends on a number of factors that characterize the related cases. In particular, accuracy is an increasing function of their autocorrelation (i.e., the correlation among attributes, and in particular the class labels, of related cases), a decreasing function of relation density, and a decreasing function of attribute predictiveness (i.e., their correlation with class label) (Jensen & Neville, 2002; Sen *et al*., 2008). As explained below, the task of object classification on cases extracted from maritime video exhibits some of these characteristics. Therefore, it is a suitable candidate for collective classification.

Two broad categories of collective classification algorithms have been studied, as distinguished by how they perform inference:

1. *Local collective inference*: These algorithms operate on a vector space representation of attributes obtained by transforming a graph of related cases. The Iterative Classification Algorithm (ICA) (Neville & Jensen, 2000), Gradual Commit (McDowell *et al*., 2007b), and Gibbs sampling (Geman & Geman, 1984) are some example techniques.
2. *Global collective inference*: These algorithms operate directly on a graph of related cases rather than attribute vectors. Examples in this category include loopy belief propagation (Pearl, 1988) and relaxation labeling (Rosenfeld *et al*., 1976).

In this paper, we apply ICA to our task. We selected it due to its simplicity, efficiency, and comparatively good performance (Sen *et al*., 2008).

Local collective classification algorithms operate on a representation of cases that includes both *intrinsic* and *relational* attributes, where the former describe properties of an individual case and the latter denote relations among cases. In this context, collective classification is a two-stage supervised learning process:

1. *Bootstrap classification*: Effective relational representations for a case typically include attributes defined as relations on the values of the class labels of related cases. For example, in the maritime domain, a relational attribute may include the distance of one track to another and the category label of the related object. However, the dependency of relational attribute values on class labels of related cases poses a problem: at the start of the classification, the class labels of the related cases are sometimes unknown, which implies that their relational attribute values cannot be computed. To jump start this process, an initial prediction for the class labels must be obtained. This is accomplished by applying a classifier to only the *intrinsic* attributes. Any conventional supervised learner (e.g., Naïve Bayes, SVMs, a case-based classifier) can be used for bootstrap classification. In this paper we use a simple case-based classifier.

2. *Collective inference*: This inherently parallel process is simulated by iterating over a loop of two steps:

    a. *Predict relational attribute values*: Based on the class labels obtained in the previous step, these values are computed to complete the case representation.

    b. *Perform local classification*: The classifier learned during the bootstrap step is used to classify cases with their predicted relational attribute values.

    Typically, the accuracy of relational attribute value predictions and local classifications increase over subsequent iterations. For the ICA algorithm, iterations of collective inference cease when there are no changes to classification predictions in successive iterations, or after a predetermined max number of iterations. Empirical evaluations of ICA show that it typically converges in a relatively small number of iterations (e.g., 10) (McDowell *et al*., 2007a).

In summary, the supervised classifier learned during the bootstrap step has access to only the (non-relational) intrinsic attributes, whereas it also has access to the relational attributes during collective inference.

In this paper, we assume no links between the training and test sets; this is known as the *out-of-sample* task (Neville & Jensen, 2005). Thus, the classifier is trained on a set of completely defined cases in step 2.b because the labels of related cases, from which the relational attribute values are derived, are all available (i.e., either given or predicted). We also assume that the relations to be used are pre-selected rather than learned. We address the implications of these and other assumptions in our evaluation in Section 5 and in the subsequent discussion in Section 6.

```
ICA (Tr,Te,NR,R,n,S)=
//Tr = Training data, Te = Test data, NR = non-relational features,
//R = rel.features, n = #iterations, S = supervised learner
Tr.R.values ←setRelFeatures(Tr,R)          //Relational value estimation
M←learnModel(Tr,NR,R,S)                    //Learn initial relation model
Te.Labels ←classify(Te,Tr,M,NR,∅)          //Bootstrap classification
for j=0 to n                               //Collective inference
  Te.R.values ←setRelFeatures(Te,R)          //Relational value estimation
  Te.Labels ←classify(Te,Tr,M,NR,R)          //Local classification
Return Te.Labels                           //Return final labels
```

Figure 2: Pseudocode for the Iterative Collective Algorithm (ICA)

4.2 Case-Based Collective Inference

In Section 4.1, we described a simple collective classifier called the Iterative Classification Algorithm. Figure 2 presents ICA's pseudocode where, in this study, we use a case-based algorithm (for *S*) to perform supervised learning and prediction.

Case-based classification predominantly involves retrieving similar cases and reusing their class labels to predict the label for a new classification problem (López de Mántaras *et al*., 2005). Below we describe our case representation for maritime object classification, followed by the retrieval and reuse methods we use. Case retention can be important in an application like ours, but we leave it for future work.

Case representation: The object classifier receives a structured representation of tracks as its input. A track comprises multiple events, each resulting from a change in an object's direction or speed. Ideally, a track represents a single moving maritime object. However, MAAW's Video Processor can make mistakes while grouping multiple events from a scene into multiple tracks. Our goal is to use the Object Classifier to reduce errors in categorizing images and use vessel category labels provided by the Object Classifier to correctly rebuild the tracks. That is, we classify objects for each event in a track instead of the track as a whole. Moreover, in our application, the track must be repeatedly classified as soon as it is detected and its classification revisited as the track unfolds. Therefore, ***we represent each event in a track as a case within MAAW.***

We use a typical <problem, label> representation for our cases. Problems are represented by intrinsic and relational attributes. *Intrinsic* attributes of a case are those attributes of a maritime object that are independent of other objects. For our task, these include the following three groups of 19 attributes (see Figure 3):

1. *Object position*: This represents the position of a maritime object in a two-dimensional coordinate system detected and extracted by the Video Processor from the maritime video. It is a tuple <p^x, p^y> comprising two continuous real values.

2. *Object velocity*: This represents the velocity vector (i.e., speed and direction) of a maritime object. Like object position, the velocity vector is represented in two dimensions using a tuple <v^x, v^y> comprising two continuous real values.

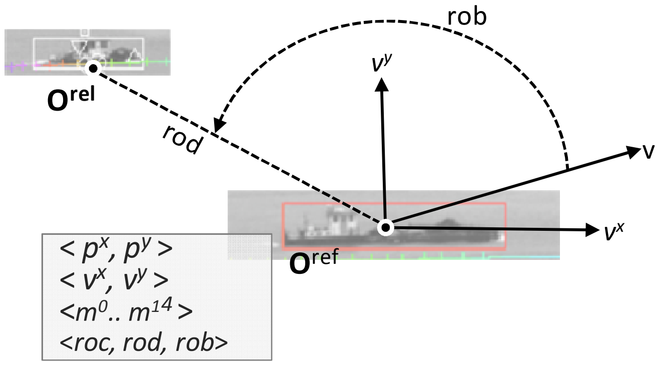


Figure 3: Attributes representing problems in cases denoting related maritime objects

3. *Object image moments*: Our Video Processor extracts images of objects from a scene including its shape, which it converts into a characteristic shape signature. Shape signatures or moments are a commonly used technique for analysis and comparison of 2D shapes. They capture information such as orientation, size, and shape boundary (Leu, 1991). We generate fourth order moments, which is a tuple comprising 15 real continuous values <m^0…m^{14}>.

In addition to these attributes, we employ the following group of *relational* attributes:

4. *Closest track object*: These three attributes encode the spatial relationship of a reference object (i.e., the object that the case represents) in a maritime scene to a related maritime object that is the closest to it. The distance between a reference object and a related object is computed based on their positions in the two-dimensional real world coordinates. The attributes comprise a tuple of three values <*roc, rod, rob*>:
    a. Related object category (*roc*): This is a categorical label of the related object selected from our Maritime Ontology.
    b. Related object distance (*rod*): This is the distance of the related object from the reference object represented by a continuous real value greater than or equal to 0. (We define our distance function below.)
    c. Related object bearing (*rob*): This is the angle between the velocity vector of the reference object and the position vector of a related object.

Other (e.g., temporal) relationships among objects exist that we could use for maritime object classification, but we leave their consideration for future study.

Case retrieval: A new problem in MAAW refers to an unlabeled object in a maritime scene. We retrieve the *k* most similar stored cases by comparing each of them with the new problem to assess their overall similarity. We compute the overall similarity by a weighted aggregation of attribute similarities, where the definitions for each of the four groups of attributes are defined as shown below.

i. *Positional similarity*: We compute the positional similarity *PosSim*(o_i, o_j) of two objects o_i and o_j as follows:

$$PosSim\ (o_i, o_j) = 1 - dist\ (o_i, o_j) \big/ MaxDist\ \ , i \neq j \qquad (1)$$

$$dist(o_i, o_j) = \sqrt{(p_i^x - p_j^x)^2 + (p_i^y - p_j^y)^2}$$

$$MaxDist = \sqrt{(\max(p^x) - \min(p^x))^2 + (\max(p^y) - \min(p^y))^2}$$

where *dist()* is the Euclidean distance between two maritime objects computed using the attributes representing their respective positions (i.e., the tuple (p^x,p^y)). MaxDist is a similarity metric parameter representing the maximum possible distance for a pair of objects, computed from their position values over the entire case base.

ii. *Velocity similarity*: We compute the similarity *SpeedSim*(o_i, o_j) of the speed of two maritime objects o_i and o_j as follows (we ignore directional differences for this task):

$$SpeedSim\ \ (o_i, o_j) = 1 - diff\ (o_i, o_j) \big/ \sigma_s, i \neq j \qquad (2)$$

$$diff\ (o_i, o_j) = \sqrt{(v_i^x - v_j^x)^2 + (v_i^y - v_j^y)^2}$$

where σ_s is a similarity metric parameter representing the standard deviation (i.e., variance) of object speeds over the entire case base.

iii. *Moment similarity*: We compute the similarity *MomSim*(o_i, o_j) of the image moments of two maritime objects o_i and o_j as follows:

$$MomSim\ (o_i, o_j) = \sum mvSim\ ^k (o_i, o_j) \big/ 15, 0 \leq k \leq 14 \qquad (3)$$

$$mvSim^k (o_i, o_j) = 1 - \min(1, \left|(mv_i^k - mv_j^k)\right| / \sigma_m^k) \qquad (3.1)$$

Equation 3 averages the similarities across 15 moment value similarities, where each moment value similarity *mvSim*^{k}(o_i,o_j) is calculated using Equation 3.1, which computes the minimum of the proportional difference of the k^{th} moment values mv^k. This metric uses 15 parameters, σ_m^k, each representing the variance of the k^{th} moment value across the entire case base.

iv. *Closest object similarity*: This metric assesses the similarity of pairs of spatially related objects. The attribute *closest track object* captures the spatial relation between a reference object and its closest related object. This metric, *ClobSim*(o_i, o_j), compares this relation in two parts (see Equation 4). First, it checks to see if the categories of related objects (i.e., roc) are the same. Then it compares the distance (i.e., *rod*) using *rdistSim()* and the bearing (i.e., *rob*) *rbearingSim*(). Equation 4 averages the distance and bearing similarities.

The distance similarity is computed using Equation 4.1, which uses a metric parameter, σ_{rod}, which represents the variation of *rod* across the entire case base. The bearing similarity is computed in four parts based on the four quadrants of a circle centered on the reference object (π/2, π, 3π/2, 2π) that roughly represent the forward, rightward, backward, and leftward topological spaces of an object.

These are forward similarity (*fsim*), backward similarity (*bsim*), rightward similarity (*rsim*) and leftward similarity (*lsim*), respectively:

$$ClobSim\,(o_i, o_j) = 0, if\,(roc_i \vee roc_j = "NONE") \vee (roc_i \neq roc_j)) \qquad (4)$$

$$= (rdistSim(o_i, o_j) + rbearingSim(o_i, o_j))/2, otherwise \qquad (4.1)$$

$$rdistSim(o_i, o_j) \;= 1 - \left| rod_i - rod_j \right| / \sigma_{rod}$$

$$rbearingSim(o_i, o_j) = \min(\min(fsim, bsim), \min(rsim, lsim))$$

for $\theta = rob_i - rob_j$

$fsim(\theta) = 2*|\pi/2 - \theta|/\pi$ when $0 < \theta < \pi/2$
$= 1- 2*|2\pi - \theta|/\pi$ when $3\pi/2 < \theta < 2\pi$
$= 0$ otherwise

$bsim(\theta) = 1- 2*|\pi/2 - \theta|/\pi$ when $\pi < \theta < 3\pi/2$
$= 1$ when $\theta = \pi$
$= 0$ otherwise

$rsim(\theta) = 1- 2*|3\pi/2 - \theta|/\pi$ when $\pi < \theta < 2\pi$
$= 1$ when $\theta = 3\pi/2$
$= 0$ otherwise

$lsim(\theta) = 1- 2*|\pi/2 - \theta|/\pi$ when $0 < \theta < \pi/2$
$= 1$ when $\theta = \pi/2$
$= 0$ otherwise

The function we use to compute aggregate similarity $Osim(o_i,o_j)$ for the learned classifier is as follows:

$$Osim(o_i,o_j) = (PosSim(o_i,o_j)+ SpeedSim(o_i,o_j)+ MomSim(o_i\ o_j))/3 \qquad (5)$$

$$Osim(o_i,o_j) = (PosSim(o_i,o_j)+ SpeedSim(o_i,o_j)+ MomSim(o_i,o_j)+ ClobSim(o_i,o_j))/4 \qquad (6)$$

where Equation 5 refers to the computation before relational values have been computed (i.e., during the bootstrap phase) and Equation 6 refers to the situation after the relational values have been computed (i.e., during collective inference). For the sake of simplicity, we ignore differential weighting of features in this paper, leaving this for future study.

Case reuse: We use the similarity-weighted voting kernel function for reusing the labels from the *k* most closely matching cases. This kernel collates the votes for the candidate category labels from each of the *k* cases, where each offers its *Osim()* value as a vote toward its object label. The kernel then computes the total vote for each candidate label by summing over all the votes it receives, and selects the label with the largest vote as the label for the new problem.

Supervised learning: Learning a case-based classifier can include learning/tuning its similarity metric from a memory of stored cases. This can involve, for example, feature weight learning and computing the values of metric parameters. In this paper, we perform only this latter task. We computed the settings of the parameters for each of the four parameters described above (i.e., MaxDist for positional similarity, σ_s for velocity similarity, σ_m^k for moment similarity, and σ_{rod} for closest object similarity). This entails estimating their value over the entire case base. For example, σ_{rod} is the standard deviation, a statistic computed over the real-valued attribute *rod*.

5 Evaluation

5.1 Objective

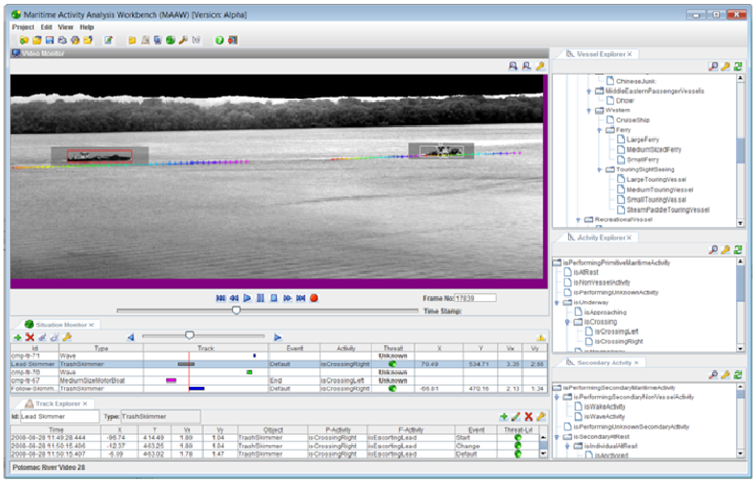
Our objective was to evaluate whether using a collective classification approach for our maritime object classification task attains a significantly higher accuracy than does a conventional supervised learning algorithm. In other words, we formulate the following null hypothesis:

H^0 There is no difference between the maritime object classification accuracy obtained by a collective classifier and the accuracy obtained by a conventional but otherwise equivalent supervised learning algorithm.

5.2 Method

Data: We selected two days of video of maritime activities on the Potomac River in Washington, DC. We used the Video Processor on this video to detect tracks of moving maritime objects and their attributes (e.g., position and velocities at different points in time). Using MAAW, we then labeled all the events in a track with appropriate object categories (see Figure 4). These object category labels were chosen from a Maritime Ontology (a taxonomy of objects, partially visible in Figure 4) that we developed using MAAW. Typically, leaf nodes of the ontology were selected, but subject matter experts were also allowed to select intermediate nodes when the object could not be visually categorized at the most specific level.

Our database included 1578 cases of labeled objects. The database included cases in 23 object categories from our Maritime Ontology, with proportions ranging from 46.64% to 0.13%. The top three most populous labels were *wave* (46.64%), *small-touring-vessel* (9.76%), and *wake* (7.41%). Half the object categories (e.g., *steam-paddle-touring-vessel*) were relatively rare and occurred less than 2% of the time in our data set.



Id	Type	Track	Event	Activity	Threat	X	Y	Vx	Vy
cmp-tr-71	Wave				Unknown				
Lead Skimmer	TrashSkimmer		Default	isCrossingRight	●	70.49	534.71	3.35	2.66
cmp-tr-75	Wave				Unknown				
cmp-tr-67	MediumSizeMotorBoat		End	isCrossingLeft	Unknown				
Follow-Skimm...	TrashSkimmer		Default	isCrossingRight	●	-56.81	470.16	2.13	1.34

Id: Lead Skimmer Type: TrashSkimmer

Time	X	Y	Vx	Vy	Object	P-Activity	F-Activity	Event	Threat-Lvl
2008-08-28 11:49:28.444	-95.74	414.49	1.89	1.04	TrashSkimmer	isCrossingRight	isEscortingLead	Start	●
2008-08-28 11:50:15.406	-12.37	463.25	1.89	1.04	TrashSkimmer	isCrossingRight	isEscortingLead	Change	●
2008-08-28 11:50:15.407	-6.09	463.92	1.78	1.47	TrashSkimmer	isCrossingRight	isEscortingLead	Default	●

Figure 4: MAAW can be used to label the extracted maritime tracks

Algorithms: We implemented two algorithms to conduct a comparative empirical evaluation. They were implemented using the Knexus Classification Workbench (KCLAW), a proprietary Java library for classification tasks:

1. ICA_0: This is a conventional case-based classifier (kNN) that does not perform collective classification. It differs from ICA in that it performs no collective inference, and does not employ relational attributes.
2. ICA: We summarized this simple collective classifier in Section 4.1 and detailed its application to our maritime object classification task in Section 4.2.

Performance Measure: We used classification accuracy as the performance measure with some modification. Given the nature of our domain, we considered graded misclassification costs based on the Maritime Ontology of object labels. In particular, we permit misclassification costs to be less than 1, depending on the taxonomic relationship between the correct and predicted labels. To do this, we used the Maritime Ontology to compute a misclassification cost matrix. For example, if a *small-motorboat* was classified as a *medium-sized-motorboat* the classification error was 0.5 rather than 1.0 because they are siblings in this taxonomy.

Test Procedure: We adopted a leave-one-out cross validation (LOOCV) test procedure with some modifications. Conventional LOOCV procedures use one case from the database for testing and the remainder for training, cycling through the entire case base and averaging the results of individual tests. We cannot use this here because collective inference operates on a *graph* of related cases, and we choose to eliminate any relations between the training and test cases. Therefore, we grouped cases that refer to co-occurring tracks and events within the same track; each such grouping yields a single fold (i.e., each fold's cases have no relations with cases in other folds). Next, we treated each fold as a test set and the union of cases from the remaining folds as the corresponding training set (i.e., the case base). This yields 177 folds, of which 77 contain cases with relational attribute values. The average number of cases per fold across the entire data set is 8.92. The average number of cases in relational folds was marginally greater (10.79). ICA_0 and ICA were applied to each test set (i.e., fold) and their classification accuracy was recorded.

Analysis: We used a paired student's 1-tailed t-test to evaluate the null hypothesis $\mathbf{H^0}$.

5.3 Results and Analysis

We compared the performance of ICA_0 and ICA under two dataset conditions:

1. *Relational Only*: To obtain insight on their true performance differences, we compared the algorithms using *only* those 77 folds that contain relations.
2. *Overall*: To assess the overall impact of collective inference (at least, as embodied in ICA) for our application, we compared the two algorithms using *all* 177 folds to obtain an aggregate performance measure.

Table 1 summarizes the results. The average classification accuracies of ICA_0 and ICA for the Relational Only condition are 46.90% and 51.85% respectively (p=0.0001). Thus, we reject our null hypothesis $\mathbf{H^0}$ and confirm that ICA, a case-based collective classifier, attains significantly higher accuracy than does an otherwise equivalent conventional (i.e., non-relational, non-collective) case-based classifier for our maritime object classification task. For the Overall condition, ICA still significantly outperforms ICA_0 (i.e., 53.23% and 56.06% respectively (p=0.0019)) although, as expected, their performance difference is smaller (4.95 vs. 2.83). There are a large number of classes in our domain and many of them occur rarely. Thus, ICA_0's classification accuracy for the Relational Only condition is substantially lower than it is for the Overall condition.

Table 1: Average Classification Accuracies for the Collective and Non-Collective Classifiers on the Maritime Object Classification Task

Comparison Scope	ICA_0	ICA	Significance
Relational Only	46.90	51.85	0.0001
Overall	53.23	56.06	0.0019

6 Discussion

Our algorithm benefited greatly from experimenting with alternative similarity functions. For example, while not reported here, we found no benefit for the collective classifier until we used a similarity metric that transformed the bearing into topological quadrants. Although we compared the performance of our algorithms using a graded (non-binary) classification error measure, our conclusions remain valid when we use a binary classification measure.

Performance could be further improved by using higher quality data and refining the collective classification algorithm. First, the data we are using is noisy; there are large variations in position detection (e.g., the position at which an object is detected can be inaccurate due to low-resolution imagery). Also, the shape geometry uses coarse techniques. We are currently addressing these issues. Also, we plan to improve tracking by providing feedback from the Behavior Interpreter to the Video Processor (see Figure 1) so as to facilitate the learning of more accurate appearance models.

Second, ICA's behavior could be improved. While we are using the *closest track object* relation, we have not yet examined alternative relations that may be more appropriate for this domain. Thus, we will study methods that can automatically identify relations, and potentially increase classification accuracy. Also, our similarity metric is primitive; performance may be improved by assigning and learning the values of attribute weights. Likewise, our collective inferencing algorithm is non-optimal. By eagerly using all the predicted labels in each iteration, if many are wrong, then classification accuracy could suffer. Accuracy may increase if we use a cautious variant of ICA (McDowell *et al.*, 2007b), which would not use low-confidence classification predictions when computing relational attribute values. Finally, collective classification accuracy can be increased by methods that can increase the data's autocorrelation (Aha, 2008), and we plan to test methods with this ability.

This paper describes our initial step towards developing a capability that can assist watchstanders with force protection monitoring tasks. We plan to evaluate our algorithm's utility on additional video of ports, harbors, and other high-traffic maritime areas. In addition, we would like to use additional sensors (e.g., 3-D cameras, infrared, long-range), and, ideally, arrange them on-board to provide 360%, real-time surveillance coverage for use in a variety of conditions (e.g., night, fog, precipitation) in many maritime environments.

7 Conclusion

Maritime surveillance for counter terrorism and force protection is manually intensive and error prone due to information overload, fatigue, and imperfect sensors. Although there is a significant opportunity for automated threat analysis from surveillance video, this problem is challenging. For example, image processing techniques may erroneously identify objects, and the low-level sensor data can be noisy.

In this paper, we focused on object recognition, an initial part of the problem of performing automated threat analysis from surveillance video. We took a unique approach to the problem by transforming a maritime scene into a graph of spatially related objects, instead of considering each object independently. This enabled us to represent and exploit the information contained in the contextual cues (i.e., the relations among objects) by applying collective classification algorithms. For one such algorithm, the Iterative Collective Algorithm (ICA), we found that it can significantly increase classification accuracy when using a case-based classifier.

We developed a novel representation for maritime object classification, applied a case-based collective classifier, and empirically demonstrated its utility. We used a domain-specific function for computing the similarity of topological relations.

There are many issues that we plan to address in our future work to improve on the methods presented here. For example, we will explore the use of cautious approaches for collective classification (McDowell *et al*., 2007b) and other more sophisticated collective inference algorithms (Sen *et al*., 2008; Aha, 2008). We will also enhance our relational representation to include temporal relations, and assess methods for automatically transforming and selecting relations for our case representation. Finally, we will investigate the use of similarity metric learning techniques.

Acknowledgements

Thanks to Ralph Hartley for implementing MAAW's image processing software. This research was supported by the Office of Naval Research and NRL.

References

Aboutalib, S., & Veloso, M. (2007). Towards using multiple cues for robust object recognition. *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 189-196). Honolulu, HI: ACM Press.

Adams, S., & Goel, A.K. (2007). A STAB at making sense of VAST data. In C. Geib & D. Pynadath (Eds.) *Plan, Activity, and Intent Recognition: Papers from the AAAI Workshop* (Technical Report WS-07-09). Vancouver (BC), Canada: AAAI Press.

Aha, D.W. (2008). Object classification in a relational world: A modest review and initial contributions. *Proceedings of the Nineteenth Irish Conference on Artificial Intelligence and Cognitive Science* (pp. 1). Cork, Ireland: Unpublished.

Burke, R., & Kass, A. (1995). Supporting learning through active retrieval of video stories. *Journal of Expert Systems with Applications*, **9**(5), 361-378.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Transactions on Pattern Analysis and Machine Intelligence,* **6**, 721-741.

Jensen, D., & Neville, J. (2002). Linkage and autocorrelation cause feature selection bias in relational learning. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 259–266). San Francisco, CA: Morgan Kaufmann.

Johnson, C., Birnbaum, L., Bareiss, R., & Hinrichs, T. (2000). War stories: Harnessing organizational memories to support task performance. *Intelligence*, **11**(1)*,* 17-31.

Jung, J., Han, I., & Suh, B. (1999). Risk analysis for electronic commerce using case-based reasoning. *International Journal of Intelligent Systems in Accounting, Finance, & Management*, **8**, 61-73.

Leu, J.-G. (1991). Computing a shape's moments from its boundary. *Pattern Recognition,* **24**(10), 949-957.

Lipton, A.J., Heartwell, C.H., Haering, N., & Madden, D. (2009). Critical asset protection, perimeter monitoring, and threat detection using automated video surveillance. Unpublished manuscript. [http://www.objectvideo.com/products/onboard/whitepapers]

López de Mantaras, R, McSherry, D., Bridge, D.G., Leake, D.B., Smyth, B., Craw, S., Faltings, B., Maher, M.L., Cox, M.T., Forbus, K.D., Keane, M., Aamodt, A., & Watson, I.D. (2005). Retrieval, reuse, revision and retention in case-based reasoning. *Knowledge Engineering Review*, **20**(3), 215-240.

MacNeil, R. (1991). Generating multimedia presentations automatically using TYRO, the constraint, case-based designer's apprentice. *Proceedings of the Workshop on Visual Languages* (pp. 74-79). Kobe, Japan: IEEE Press.

McDowell, L.K., Gupta, K.M., & Aha, D.W. (2007a). Case-based collective classification. In *Proceedings of the Twentieth International FLAIRS Conference*. Key West, FL: AAAI.

McDowell, L., Gupta, K. M., and Aha, D.W. (2007b). Cautious inference in collective classification. *Proceedings of the Twenty-Second Conference on Artificial Intelligence* (pp. 596-601). Vancouver, Canada: AAAI Press.

Micarelli, A., & Sansonetti, G. (2007). Case-based anomaly detection. *Proceedings of the Seventh International Conference on Case-Based Reasoning* (pp. 269-283). Belfast, Northern Ireland: Springer.

Murdock, J.W., Aha, D.W., & Breslow, L.A. (2003). Assessing elaborated hypotheses: An interpretive case-based reasoning approach. *Proceedings of the Fifth International Conference on Case-Based Reasoning* (pp. 332-346). Trondheim, Norway: Springer.

Neville, J., & Jensen, D. (2000). Iterative classification in relational data. In L. Getoor and D. Jensen (Eds.) *Learning Statistical Models from Relational Data: Papers from the AAAI Workshop* (Technical Report WS-00-06). Austin, TX: AAAI Press.

Neville, J., & Jensen, D. (2005). Leveraging relational autocorrelation with latent group models. *Proceedings of the Fifth International Conference on Data Mining* (pp. 322-329). Houston, TX: IEEE Press.

ObjectVideo (2009). Intelligent video surveillance increases security at seaports. [http://objectvideo.com]

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, CA: Morgan Kaufman.

Perner, P. (1993). Case-based reasoning for image interpretation in non-destructive testing, Proceedings of the First European Workshop on Case-Based Reasoning, Vol. II, (pp. 403-409) Kaiserslautern, Germany: University of Kaiserslautern.

Perner, P., Holt, A., & Richter, M. (2005). Image processing in case-based reasoning. *Knowledge Engineering Review*, **20**(3), 311-314.

Rhodes, B.J., Bomberger, N.A., Seibert, M., & Waxman, A.M. (2005). Maritime situation monitoring and awareness using learning mechanisms. *Proceedings of Situation Management: Papers from the Military Communications Conf.* Atlantic City, NJ: IEEE.

Rincón, M., & Martínez-Cantos, J. (2007). An annotation tool for video understanding. *Proceedings of the Eleventh International Conference on Computer Aided Systems Theory* (pp. 701-708). Las Palmas de Gran Canaria, Spain: Springer.

Rosenfeld, A., Hummel, R.A., & Zucker, S.W. (1976). Scene labeling by relaxation operations. *Transactions on Systems, Man, and Cybernetics*, **6**(6), 420-433.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, **29**(3), 93-106.

Zhang, D., & Nunamaker, J.F. (2004). A natural language approach to content-based video indexing and retrieval for interactive e-learning. *Transactions on Multimedia*, **6**(3), 450-458.