ANALYTiC: An Active Learning System for Trajectory Classification

A. Soares[*], C. Renso [†], and S. Matwin [‡]

Abstract

There is an increasing amount of trajectories data becoming available by the tracking of various moving objects, like animals, vessels, vehicles and humans. However, these large collections of movement data lack semantic annotations, since they are typically done by domain experts in a time consuming activity. A promising approach is the use of machine learning algorithms to try to infer semantic annotations from the trajectories by learning from sets of labeled data. This paper experiments active learning, a machine learning approach minimizing the set of trajectories to be annotated while preserving good performance measures. We test some active learning strategies with three different trajectories datasets with the objective of evaluating how this technique may limit the human effort required for the learning task. We support the annotation task by providing the ANALYTiC platform, a web-based interactive tool to visually assist the user in the active learning process over trajectory data.

Keywords: Active Learning; Semantic Annotation; Trajectory Classification

1 Introduction

We are witnessing a rapid increase in the use of positioning devices, from new generation smart-phones to GPS-enabled cameras, sensors, and indoor positioning devices. Thanks to the fact that these devices are becoming smaller and cheaper, many kinds of objects are nowadays tracked, like vehicles, vessels, animals, and humans. This results in huge volumes of spatio-temporal data which require dedicated methods to properly analyze them. In this context, there is a growing interest in the semantic enrichment of movement data: many application fields benefit from the synergic combination of pure geometrical spatio-temporal data with semantic information, denoted as *semantic* or *annotated trajectories*. Tourism, cultural heritage, traffic management, animal behavior or vehicle tracking, are just a few examples of studies benefiting from annotated trajectories [14]. However, methods to make explicit the semantic dimension of movement data are still lacking and finding methods to automatically or semi-automatically infer trajectory labels in large datasets is an ongoing challenge [10].

Machine learning is a promising direction for annotating trajectories with semantic labels by iteratively learning from labeled training sets (training sets) to build models (or classifiers) that are then applied to unlabeled data to obtain a labeled dataset with a given accuracy. Machine learning is extensively used in many prediction-based applications, where the predictions are purely numeric (e.g., velocity) or class based (e.g. low or high velocity), and where they can learn from large labeled training sets. In the case of the trajectory domain, when the inferred classes are semantic based (e.g. kind of transportation or activity performed), the availability of training data depends mainly on trajectories manually annotated by humans. These annotations are, however, difficult to obtain since they are extremely time-consuming for the domain expert who needs to annotate large trajectories datasets correctly. The question, therefore, is: *is it possible to annotate trajectories automatically by analyzing their features, thus reducing the human effort involved in manually annotating them?*

[*]Dalhousie University, Institute for Big Data Analytics, Halifax, Canada. Email: amilcar.soares@dal.ca

[†]ISTI-CNR, Pisa, Italy. Email: chiara.renso@isti.cnr.it

[‡]Dalhousie University, Department of Computer Science, Halifax, Canada and Institute for Computer Science, Polish Academy of Sciences, Warsaw. Email: stan@cs.dal.ca

However, a good performance generally requires that the training set is large, demanding a substantial effort from the domain expert to provide a sufficient number of examples to the classifier. We, therefore, reformulate the above question into three distinct research questions:

RQ1 - Is there a machine learning method that will allow an automatic trajectory classification by minimizing the number of required human labeled trajectories?

Active learning is a machine learning technique that learns a model in an interactive way by selecting the instances where the classifier is the most uncertain. Therefore, active learning could be an effective approach to reducing human labeling effort [18]. However, to the best of our knowledge active learning has not been experimentally applied to trajectory data so far and many questions remain unanswered such as:

RQ2 - Is this machine learning method effective for trajectory data?

RQ3 - How can the user be assisted in labeling trajectories?

The contribution of this paper in answering the above questions is twofold: first, we experimentally apply active learning in the trajectory domain for the semantic labeling of movement data. Trajectories are classified into predefined classes and these classes become the semantic labels of the trajectories. We evaluate the efficacy of the approach and discuss which active learning strategy performs better. Specifically, we provide an empirical analysis of the active learning strategies known as uncertainty sampling (UNC) and query-by-committee (QBC), applied to three trajectories datasets in different domains. We show how these techniques obtained high performances with a small number of trajectories for training, reducing as a consequence the labeling effort. Second, we propose a web-based visual interactive annotation tool named ANALYTiC (AN Active Learning sYstem Trajectory Classification) supporting the user in the active learning trajectory annotation task. By designing an effective visual support we can enable fast, accurate, and hopefully trustworthy semantic annotation of the learning training set. The basic idea of active learning is to interactively submit to the annotator the best set of trajectories to annotate to improve the classifiers' performance. In machine learning, having a good sample of labeled trajectories is essential to reach good performance values, thus the ANALYTiC tool is a step towards this direction.

To the best of our knowledge, this is the first attempt to exploit active learning techniques in the trajectory semantic classification field. At the same time, this is the first annotation tool for trajectories tailored to the active learning task. We think that this work could pave the way to a better use of machine learning for trajectory semantic enrichment.

This paper is organized as follows. The related work is briefly discussed in Section 2. Section 3 shows concepts and terminologies regarding the trajectory domain and active learning. The experimental results are presented and discussed in Section 4 while in Section 5 the ANALYTiC tool is introduced. Finally, the conclusions are shown in Section 6.

2 Related Work

Active learning is a lively research area [19], where the basic idea is to identify small but meaningful subsets of data to be labeled by an "oracle" as an input for a machine learning classifier. Active learning is gradually applied to several domains: recommendation systems (e.g. [17]), natural language (e.g. [9]), bioscience (e.g. [13]), geographical object matching (e.g. [21]) . However, to the best of our knowledge active learning has not been experimentally applied to trajectory data. Therefore, this paper offers a first exploratory study, supported by an interactive visual annotation tool, to better understand the benefits of active learning in the semantic labeling of movement data.

One common way to annotate trajectory data is to detect interesting trajectory segments (e.g. stop and moves) and enrich them with labels like the Points of Interest (see for example [2]) or more complex forms of data like Linked Open Data, as discussed in [4, 16]. These methods use some predefined criteria (like the distance from the trajectory points to a geographical object) to associate a given entity (e.g. the Point of Interest) to a trajectory. However, in many cases the trajectory (or segment of a trajectory) annotation is not related to a nearby geographical place; thus a simple distance measure is not sufficient to capture the correct annotation. This is the case for the activities of a fishing vessel at sea or the behavior of an animal moving within its habitat. In these cases we need a human being to annotate the data manually with the correct semantic label. An example of a tool supporting the manual annotation of human trajectories has been presented in [15]. Here the idea is that the user uploads his/her tracks into the system and a friendly

web-based visual interface allows him/her to browse the collected movements, display them on a map and add labels like the transportation means used, and the Points of Interest visited. In this case, there is no machine learning phase as the whole dataset of trajectories has to be manually annotated to produce a semantically labeled trajectory dataset.

Machine learning techniques have been applied to classify trajectories in domains such as maritime vessels [3], traffic data [5], and human mobility [8, 24, 22]. The paucity of labeled instances in these applications and the difficulty of labeling is often a bottleneck. In contrast to these approaches, the present paper proposes to annotate only the high-yield samples (from the classifier's performance perspective) of the whole dataset with the objective of inferring the remaining labels through a machine learning task. Besides, ANALYTiC also provides full support for the interactive active learning process on movement data. In contrast to Visual Analytics approaches, proposing visualization methods for processing trajectory data as a support for the whole analytical process [20, 6], here the visual approach is for user support in the annotation task and for driving the steps of the active learning process. However, we believe that this experiment shows the potential of combining machine learning with visual methods in the context of movement data and can be a first step in this direction.

3 Background

In this section, we introduce some concepts and terms used in the paper. We also introduce, and briefly discuss, the active learning basic terminology.

3.1 Trajectory concepts and terminologies

A *trajectory* is a list of spatio-temporal points $\tau_N = \{tp_0, tp_1, \ldots, tp_N\}$, where $tp_i = (x_i, y_i, t_i, \omega_i)$ and x_i, y_i, t_i are the spatio-temporal coordinates of the point and ω_i is a point feature: any kind of numeric information that can be extracted from a trajectory and associated with a spatio-temporal point, like the speed or direction. The point features can be acquired by a geolocation device (e.g., the moving object's instantaneous speed) or calculated using the trajectory sample (e.g., the moving object's direction variation between two consecutive points) and it is assigned to a single point.

A *trajectory feature* is any numeric information computed using the information of the entire trajectory (e.g., average or maximal speed). The difference between a point feature and a trajectory feature is that, while the former is static, the latter is more dynamic. It means that, for point features, once the information is collected or computed for a single point of a trajectory, this information will not change over time. However, the trajectory features depend on the trajectory definition. By modifying it, the feature value will have to be recomputed.

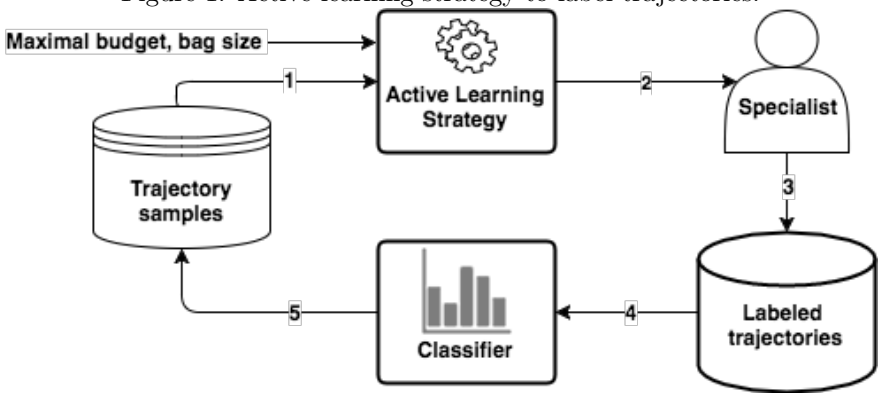
A *semantic label* (or semantic annotation) is any additional semantic and/or contextual information that can be added to a trajectory[10]. Such information can be, for example, an activity (e.g., walking, studying, driving, fishing) or a behavioral pattern (e.g., foraging or running from a predator). Henceforth, the term label refers to a *semantic label*.

3.2 Active learning strategies

According to [18], Active Learning (AL) is used in situations where obtaining labeled data is expensive or time-consuming; by sequentially identifying which examples are most likely to be useful, an active learner can sometimes achieve good performance, using far less training data than would otherwise be required. A common active learning scenario is called *pool-based sampling* [7]. This scenario assumes a small set of labeled data $\mathcal{L}$ and another pool of unlabeled data $\mathcal{U}$, where queries are selectively drawn from $\mathcal{U}$. The instances are queried according to an informativeness measure used to evaluate all instances in the pool. Assuming an annotation budget $\mathcal{B}$ and a classifier's performance measure, the active learning strategy aims to select a problem instance to be labeled by an oracle (i.e. the human annotator) in order to expand $\mathcal{L}$ and to maximize the classifier's performance measure subject to the budget constraints.

According to the research reported in [12], *uncertainty sampling* (UNC) and *query-by-committee* (QBC) are the two most frequently utilized AL strategies in the literature. In this work, UNC and QBC are compared with the most common baseline that is used for comparing with active learning strategies, named Random

Figure 1: Active learning strategy to label trajectories.



Sampling (RND). The RND strategy selects instances randomly from $\mathcal{U}$, without considering whether it provides any additional information to the classifier.

In UNC, the oracle (i.e. user) labels the instances for which the label is most uncertain. Equation 1 defines the uncertainty sampling based on conditional error, where $P_{\mathcal{L}}(y|x)$ represents the conditional probability distribution learned by the underlying classifier using the labeled set $\mathcal{L}$. The conditional error is the standard measure for probabilistic classifiers.

$$x^*_{UNC} = \underset{x \in \mathcal{U}}{\operatorname{argmax}}(1 - \max_{y \in Y} P_{\mathcal{L}}(y|x)) \tag{1}$$

The QBC approach involves maintaining a committee of models which are trained with $\mathcal{L}$, but which represent competing hypotheses. In this work, the committee of models is built with the same classifier trained with different randomly sampled initial instances. Each member of the committee votes on the labels of the queried instances. The most informative instance is the one where the committee most disagrees. Equation 2 shows the vote entropy, where y ranges over all possible class labels in Y, $V(y)$ is the number of votes that a class label receives from the committee members, and C is the committee size.

$$x^*_{QBC} = \underset{x \in \mathcal{U}}{\operatorname{argmax}} - \sum_{y \in Y} \frac{V(y)}{C} log \frac{V(y)}{C} \tag{2}$$

It is important to point out that any classifier can be combined with an AL strategy.

4 Experimenting Active Learning on Trajectory Data

As mentioned above, the objective of the active learning (AL) strategy is to select, among a set of trajectories, the ones that, when manually annotated by the user, could improve the performance of the chosen classifier.

The active learning process for trajectories is summarized in Figure 1. The process starts with a set of trajectory samples ($\mathcal{U}$). In step (1) we fix a maximal budget $\mathcal{B}$ (i.e. a maximal amount of trajectories to be queried) and a bag size (the groups of trajectories to be labeled each step) for the AL strategy. The AL strategy sub-samples a number of trajectories (step 2) defined in the bag size based on some predefined strategy (UNC, QBC, and RND) and shows the result to the human annotator (i.e. the domain specialist). The annotator analyses each trajectory evaluating the point and trajectory features and labels the trajectory (step 3). In the following step, the newly labeled trajectories are added to the training set (step 4). The classifier builds a model that is exploited to annotate the non-labeled trajectories in the pool (step 5). The procedure (steps 1 to 5) is repeated with a new bag of instances until the maximal budget of instances $\mathcal{B}$ is reached.

4.1 Trajectory datasets, methodology and evaluation metrics

We experimented with active learning in a total of three trajectory datasets covering different domains: animals, vessels and human movements. These trajectories are associated with a semantic label and therefore they act as a ground truth to evaluate the machine learning classifier's performance.

We decided to use trajectory datasets with different characteristics regarding the number of trajectories, the number of features and the balance between the classes. This allowed us to evaluate the performance of the AL strategies under different conditions. The three datasets used in this work are described below.

The first dataset is provided by the Starkey Project[1] and contains trajectories of elk, deer, and cattle. Trajectories of each animal were generated on a daily basis and the animal type was used as the class labels. The speed in m/s, the direction variation in *degrees* and the traveled distance from the previous point in *meters* were computed as the point features. Six other numerical features, provided in the dataset were used and associated with each point: soil depth, distance to nearest water supply, distance to nearest water supply from within an ungulate-proof, elevation, canopy closure of all trees nearby the animal and percent slope.

The second dataset contains data regarding vessels navigating on the Brazil's northeast coast. Each point contains information regarding fishing activities and their trajectories are labeled with the kind of activity performed: *fishing* and *not fishing*. In this dataset, the trajectories were created using the sequences of the same kind of activity performed by the same vessel. For each trajectory point, the speed in m/s, the direction variation in *degrees* and the traveled distance from the previous point in *meters* were computed as point features.

The third and last trajectory dataset is the Geolife [23], and contains data regarding human movements collected in Beijing (China). In this dataset, we considered eight different types of transportation means: walking, running, car, motorcycle, taxi, bus, train, and subway. We reduced the number of classes to three by grouping the original ones in: (i) *no vehicles* (e.g. walk or run); (ii) *rail-based* (e.g. train and subway); and (iii) *vehicles* (e.g. car, motorcycle, taxi or bus). In this dataset, the trajectories were created using the sequences of the same kind of transportation means used by the same moving object. For each point in the trajectories the speed in m/s and the acceleration in m/s^2 was computed.

For each trajectory, the minimum, maximum and average values of all point features were computed to be used as the classifier's features (i.e. trajectory features). This means that for each point feature considered for each dataset, three trajectory features were derived. Table 1 gives an overview of the characteristics of each dataset, where the balance between classes, the number of trajectories, and number of point and trajectory features can be seen.

Table 1: Dataset summary.

Dataset	**Classes names (proportion by class)**	**N. trajectories**	**N. point features**	**N. trajectory features**
Animals	Elk(55%), Deer(26.9%) and Cattle(18.1%)	25884	9	27
Fishing Vessels	Fishing(50.4%) and not Fishing(49.6%)	127	3	9
Geolife	No vehicles(48.9%), vehicles(40.2%) and Rail-based(10.9%) speeds	16515	2	6

In this work, we tested the versions of UNC introduced in [7] and QBC introduced in [1]. The random approach RND was used as a baseline. We chose the most common classifiers in the literature from the *sklearn* package [11]: (i) Logistic Regression (LR); (ii) Gaussian Naive Bayes (GNB); (iii) Decision Trees; (iv) K-Nearest (KNN); (v) Ada Boost (AB); (vi) Random Forest (RF).

We compare the three strategies using two evaluation measures: accuracy (ACC) and the area under the curve (AUC). Accuracy measures the percentage of instances that are predicted correctly, and therefore gives the quality of the classification performance. The AUC measures the probability that the classifier

[1]The Starkey Project http://www.fs.fed.us/pnw/starkey/index.shtml

will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Although accuracy is the most frequent classification performance measure, it does not consider the imbalance among the classes, a characteristic that is considered by the area under the curve (AUC) measure. Therefore we decided to use both measures in our experiments.

It is important to point out that in this work we performed only a two-class analysis of the AL strategies. We created sub-datasets for the datasets with more than two classes to compute accuracy and AUC. We recall that these measures assume positive and negative classes; therefore, we derived new datasets assuming one class as the positive class and the remaining others globally as the negative class. For example, in the case of the animals dataset, a sub-dataset was created with the elk trajectories as being the positive class and deer and cattle trajectories as the negative class. This procedure was repeated for each class in these datasets, bringing a total of 7 sub-datasets used in the experiments.

For each experiment, the train split was used as the unlabeled pool $\mathcal{U}$ and 20 instances (ten from each class) were used as the initially labeled set $\mathcal{L}$. This amount of 20 instances was found by verifying the minimum amount of instances that made all classifiers reach at least 50% of accuracy and 0.5 of AUC. For all sub-datasets, we performed a five-fold cross validation to verify how the model generalizes its results regarding an independent dataset (i.e. the test split). We also performed ten different trials using different seeds, with values 1 to 10, to initially randomly select the labeled instances. This gives a total of fifty (i.e. 5-fold multiplied by ten trials) different starting conditions in order to evaluate the performance of the AL strategies. The final values for accuracy and AUC were defined by averaging these fifty trials in all the experiments reported in this work. At each iteration, we picked the top 10 utility instances (i.e. bag size), in the case of the fishing vessels dataset, and 20 utility instances, in the case of the other datasets, ranked by an AL strategy until the maximal budged was reached. The decision of the bag size value of 10/20 has two main reasons. First, previous experiments showed that adding 1 or 5 instances at each AL iteration has a low impact on the classifier's performance (e.g. more or less than 1% and 0.01 of accuracy and AUC, respectively). Second, by using a lower value than 10/20 creates a bottleneck in the back-end of the application since many transactions would be sent to the server side. The maximal budget value of 400 was chosen based on a previous verification that showed that the classifiers combined with the AL strategies did not improve more than 1% by raising the number of trajectories for training. In the case of the fishing vessels dataset, this budget was fixed as 60 instances since only 127 instances were available.

As an answer to research question *RQ1 - Is there a machine learning method that will allow an automatic trajectory classification by minimizing the number of required human labeled trajectories?* we proposed using active learning methodology. To verify that active learning can actually reduce the number of labeled trajectories we ran a number of experiments discussed in Section 4.2.1. The answer to the research question *RQ2 - Is this machine learning method effective for trajectory data?* is discussed in Section 4.2.2.

4.2 Results and Discussions

We compared the results of the experiments with the three AL strategies using the six different classifiers in two ways. First, we investigated which AL strategy among the three was able to reach a target value for each measure determined by a five-fold cross-validation in the entire sub-dataset. The intention behind this experiment was to evaluate which AL strategy most decreases the number of instances necessary for a classifier to label the trajectories with the same value as using the entire dataset as the training data. Second, we computed the statistical difference significance between UNC, QBC, and RND. The objective of this experiment was to compute the difference in the performance between the three strategies and evaluate if this difference is statistically significant to conclude whether one is better than other. We measured the statistical difference significance through a *paired t-test*, where the pairs are the learning curves of the strategies using the sequence of budgets given to the strategies. This is detailed in Section 4.2.2.

4.2.1 Budget to target comparison

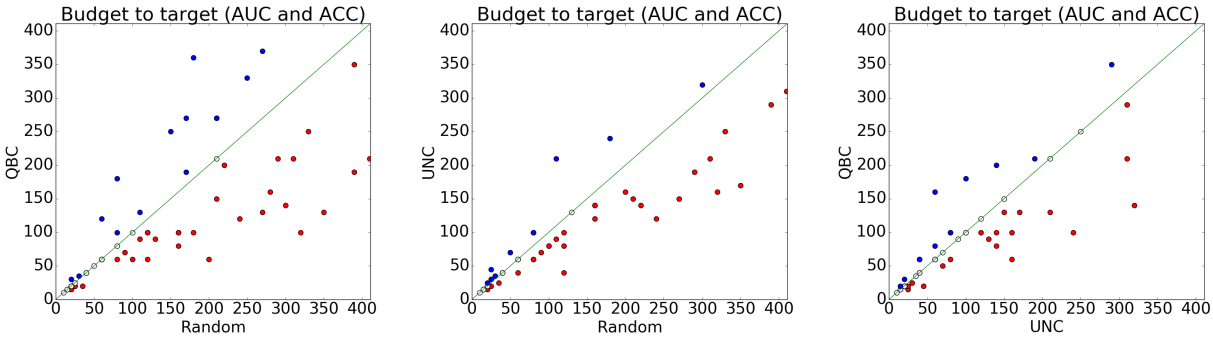
We discussed above the use of active learning, and here we discuss how the AL strategies effectively reduce the number of trajectories to be labeled. The result is summarized in Figure 2. This figure shows plots of the AL strategy which reached the target value for each measure first. First we individually compare each strategy with the RND baseline (Figure 2(a) and (b)), then we directly compare the two AL strategies UNC

and QBC (Figure 2(c)).

Each data point in Figure 2 is a pair $[x_i, y_i]$ of budget values necessary to reach the target for two different AL strategies. The total of comparisons in each figure (X_N and Y_N, where $i = 0, 1, 2, ..., N$) is equal to 84 (7 sub-datasets x 6 classifiers x 2 performance measures). The interpretation of Figure 2 is given below. When a data point intersects the $x = y$ line (i.e. green line in Figure 2), this means that the two AL strategies x and y need the same budget to reach the target value for a measure, so they are essentially equivalent. When a data point is below the line $x = y$ (i.e. red dot in Figure 2), this means that the AL strategy in the x-axis needs a higher budget to reach the target and it is, therefore, worse. Conversely, when a data point is above the line $x = y$ (i.e. blue dot in Figure 2), the AL strategy in the x-axis needs a lower budget to reach the target, and it is, therefore, better.

Since most of the experiments almost reached the target, and a difference of 2% and 0.02 for accuracy and AUC is not significant to conclude that a classifier performed better than other, we decreased the target values to evaluate better the differences obtained by the AL strategies. By reducing the target values, the QBC did not find the target only 10 times, UNC 25 times and the RND 16 times.

Figure 2: Budget to target comparison for all active learning strategies.



(a) Comparison between QBC and RND for both AUC and ACC.

(b) Comparison between UNC and RND for both AUC and ACC.

(c) Comparison between QBC and UNC for both AUC and ACC.

In the first comparison, between QBC and RND (Figure 2 (a)), we observe that QBC found the target in 38 experiments with fewer labeled instances than RND (red dots), while RND found the target earlier only 15 times than QBC (blue dots). In 22 times a tie (points on the green line) occurred (white dots). The second comparison is between UNC and the RND baseline, shown in Figure 2 (b). Here we see that UNC found the target 32 times with fewer instances than the RND, while RND found the target earlier in 23 experiments. For 17 experiments a tie occurred. The last comparison is between the two active learning strategies QBC and UNC (Figure 2 (c)). We see that QBC found the target 36 times earlier than the UNC, while UNC found the target 14 times earlier than QBC and for 26 times, a tie occurred. Another interesting finding is that a Decision Tree combined with the QBC strategy finds the budget value of accuracy and AUC earlier than RND and UNC 26 times in 28 experiments. The results also showed that when Gaussian Naive Bayes is combined with UNC, the budget value was found earlier by UNC 11 times in 14 experiments when compared with RND strategy.

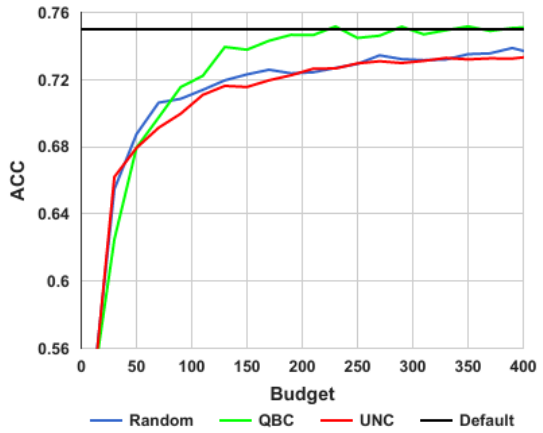
In summary, we conclude that QBC and UNC finds the target earlier than the RND baseline and QBC finds the target even sooner than UNC. This experiment answers RQ1 since we show that active learning is effective in reducing the number of trajectories to be labeled by the user. The reason for this conclusion is that QBC, using at maximum 400 instances, achieved at least the same performance value for a classifier using all data as training in 88% (74 of 84 experiments) of the experiments.

4.2.2 Learning curves comparison

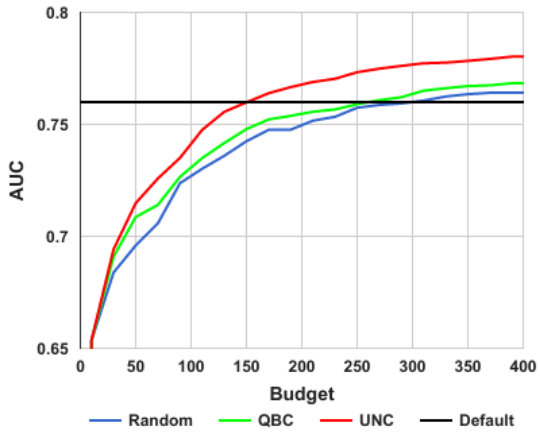
Having shown that the active learning strategies are substantially better than non-active learning, here we will go deeper into this analysis and try to understand if the difference in the learning curve is significant. The significant difference between the strategies is measured using a *p-value* of 0.05 for the *paired* $t-test$.

When the learning curve of a strategy is statistically significantly better than another, then we say this is a Win (W); when it is significantly worse than another, we say it is a Loss (L). When the differences are not significant, we say it is a Tie (T). Figure 3 shows two examples of learning curves comparisons for the three strategies. The line in black represents the results of a default configuration obtained by a five-fold cross-validation on the entire sub-dataset. Figure 3(a) shows the learning curve for the animals dataset with the cattle trajectories as the positive class and the accuracy as the measure analyzed. Here, a win (W) was assigned to QBC versus RND and QBC versus UNC while a tie (T) is assigned to UNC versus RND. Therefore in this specific case we see how QBC performs significantly better than the baseline and the other strategy, while UNC and RND are mainly in a tie. The plot also shows that a budget of 200 instances using the QBC strategy reaches the default configuration of the DT classifier. Figure 3(b) shows the AUC for the Geolife dataset with the vehicles transportation mean as the positive class. Here we observe a win (W) for UNC versus RND, a loss (L) for QBC versus UNC and a tie (T)for QBC and RND. So in the Geolife dataset, we see how the UNC is more significant than the other strategies. The figure shows that a budget of 150 instances and the UNC strategy reaches the default classifier, and with a budget of 400 it increases the AUC in more than 0.02 of AUC.

Figure 3: Learning curves comparison for two datasets.



(a) Learning curve for ACC of DT classifier in the animals(cattle) dataset.



(b) Learning curve for AUC of LR classifier for the geolife(vehicles) dataset.

The details of the results of all experiments are shown in Table 2. In each *cell* we report the number of Wins, Losses and Ties respectively. Notice that the sum of all Win, Ties and Losses is always 7, the number of sub-datasets tested with an AL strategy. The total number of experiments for all sub-datasets, each pair of compared AL strategies (QBC/RND, UNC/RND and QBC/UNC) is 42 (7 sub-datasets x 6 classifiers).

Table 2: Results

Classifier	**ACC**			**AUC**		
	QBC/RND W/T/L	**UNC/RND W/T/L**	**QBC/UNC W/T/L**	**QBC/RND W/T/L**	**UNC/RND W/T/L**	**QBC/UNC W/T/L**
Ada Boost	7/0/0	5/0/2	6/1/0	7/0/0	1/3/3	6/1/0
Decision Tree	5/2/0	2/4/1	6/1/0	4/3/0	2/3/2	6/1/0
Gaussian NB	3/3/1	6/0/1	1/0/6	0/1/6	0/0/7	7/0/0
KNN	6/0/1	6/1/0	0/5/2	1/5/1	4/2/1	0/2/5
L. Regression	5/2/0	3/1/3	4/2/1	5/2/0	1/2/4	6/0/1
R. Forest	6/1/0	7/0/0	0/6/1	3/3/1	4/3/0	0/3/4
Total	32/8/2	29/6/7	17/15/10	19/14/9	12/13/17	22/7/13

In summary, these results show that regarding statistical significance, QBC outperforms RND in accuracy for all datasets, totaling 32 wins in a total of 42 experiments. When the AUC is considered, QBC wins in

19 experiments. The comparison between UNC and the baseline shows that for accuracy, UNC outperforms RND in 29 experiments and loses only in 7, and, therefore, it is significantly better than the baseline. When the AUC is considered instead, we see that the UNC loses 17 times against the RND, ties 13 times and wins 12 times. This shows that UNC improves accuracy when compared to RND but loses in AUC. However, it can be observed that most of the losses occur when the Gaussian Naive Bayes and Logistic Regression classifiers are used. Finally, when QBC and UNC are compared, we see that QBC wins in accuracy (19 times) and wins in AUC (22 wins).

We have empirically showed that active learning (both QBC and UNC strategies) is suited to improving accuracy compared to the RND baseline in trajectory datasets. When the AUC is considered, QBC showed better results when compared to RND. Finally, the comparison between QBC and UNC showed that the first improves both accuracy and AUC.

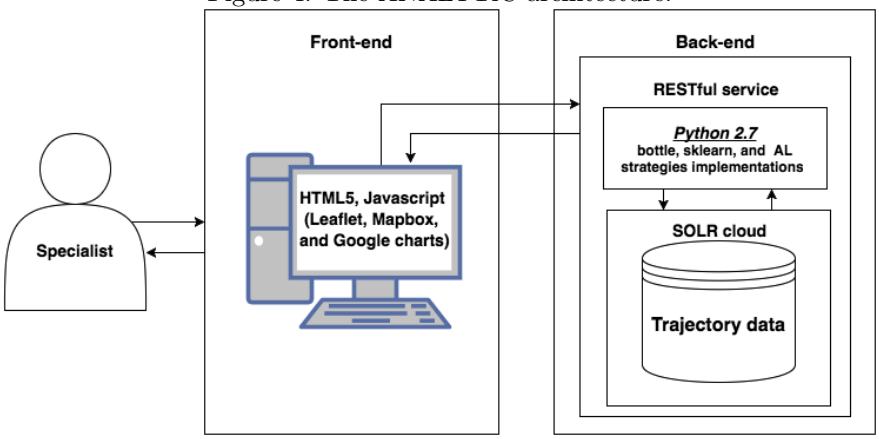
5 The ANALYTiC tool

In this section, we propose an answer to research question *RQ3 - How can the user be assisted in labeling trajectories?* The ANALYTiC visual interactive system is a possible answer to this question. The architecture is shown in Figure 4.

The system is composed of a front end interacting with the user and a back end that stores the trajectories and executes the classification algorithms and AL strategies. Since it is a web tool the user interface components were coded in HTML 5 and all the interactions with the user were coded in Javascript programming language. It is important to cite three libraries that were used in this project: for coding the map visualizations and interactions we used the mapbox[2] and leaflet[3] libraries. For showing the data as charts to the user, we used the Google charts[4] library.

The back end of the application stores the trajectory data in a SOLR[5] cloud. The main reason to use SOLR is that it is one of fastest searching platforms currently available. On the top of SOLR, we created a RESTful (REpresentational State Transfer) web service aiming to facilitate the communication between the front end and the back end. The decision to use a RESTful web-service in the back end was because it is a well-known architecture that provides interoperability between computer systems on the Internet. All the Active Learning algorithms (e.g. our own implementations), the classifiers (sklearn [11]) and data queries (SOLR) were implemented in the Python programming language (version 2.7). To provide all this infrastructure as a RESTful service we used the bottle[6] library.

Figure 4: The ANALYTiC architecture.



In the following we focus on the front end presenting the solutions to support the user interactions for

[2]The Mapbox Project - https://www.mapbox.com
[3]The Leaflet Project - https://leafletjs.com
[4]The Google Charts Project - https://developers.google.com/chart/
[5]The Apache Solr Project - http://lucene.apache.org/solr/
[6]Bottle: Python Web Framework - http://bottlepy.org/docs/dev/

the trajectory labeling and the support of the Active Learning process. An overview of the elements of the ANALYTiC front end user interface is presented in Figure 5. In Figure 5(a) we show the top panels that compose the web interface, while Figure 5(b) shows the bottom panels. In the top navigational bar of the tool (Figure 5(a)), the user can choose the dataset to annotate. The user then chooses a classifier among the six available and one of the three AL strategies (UNC, QBC or RDN).

The largest part of the interface is devoted to the actual annotation task, managed by the middle navigational bar that is composed of two panels: (i) on the left, a map displays the trajectories to be labeled; (ii) on the right, two panels show the features values (point and trajectory) useful to evaluate the trajectory label.

In a bottom part of the trajectory features we have a section where the user may label the trajectory with the most appropriate annotation and move to the next trajectory. The bottom navigational bar, shown in Figure 5(b), drives the user through the AL process (Figure 1) and displays the configuration parameters for the active learning task. This section of the interface is composed of four panels: (i) on the top left, the panel interactively displays the next action to take, thus driving the user through the steps of the active learning process; (ii) the panel on the top right summarizes the number of trajectories already labeled and the number of trajectories to be labeled to reach the total budget that was selected by the user; (iii) the panel on the bottom left summarizes the user's choices in terms of the selected classifier, the AL strategy, the bag size and maximal budget; and finally, on the bottom right, (iv) a panel manages the manipulation of the labels including the addition of new labels, the selection of the color of the label on the map, and other display features like the trajectories' weight and visibility.

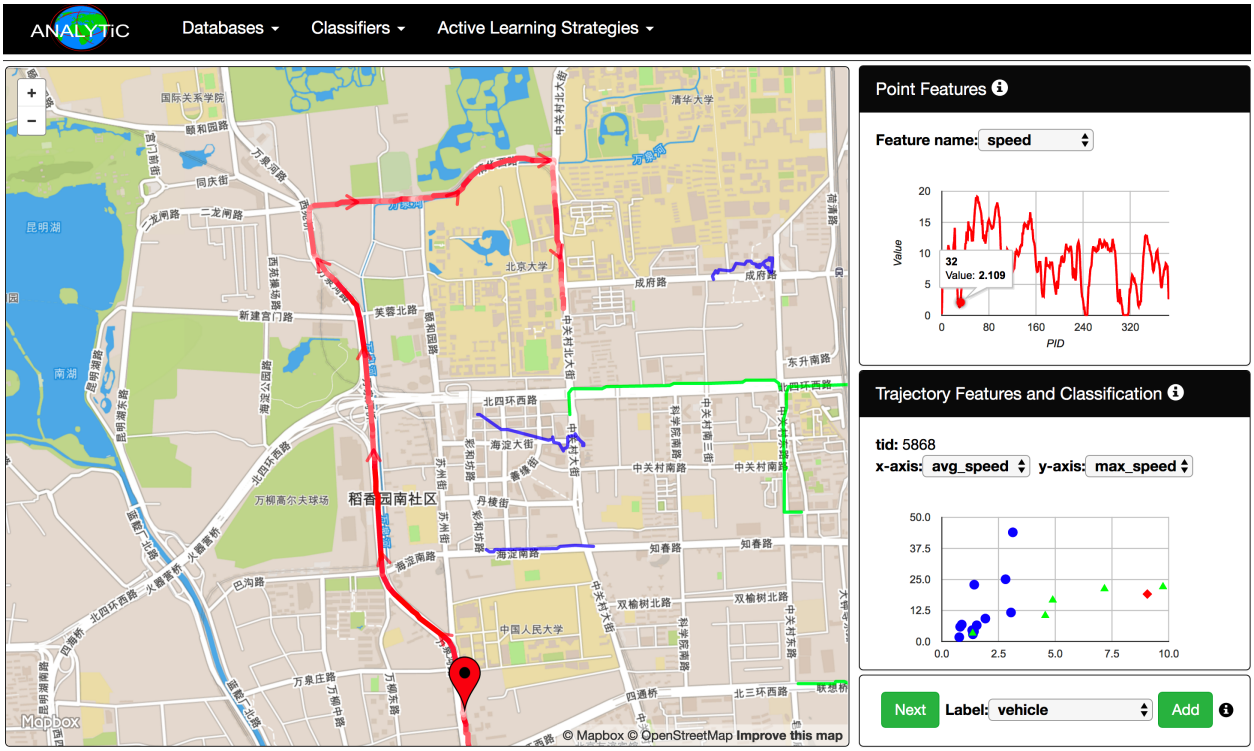
The whole active learning process is summarized as follows. The user starts the interaction by selecting the dataset to be annotated, then the classification algorithm (i.e. Logistic Regression, decision trees, etc) and the active learning strategy (Random, UNC or QBC). These selections appear in the bottom left box. Next, the user can choose the total budget of trajectories and bag size that will be used in the AL process. Having done that, the "Next" panel invites the user to the following step, namely adding the labels, in the bottom right box. From here the user may personalize the visual features like color line weight and visibility of the lines. Then the "Next Step" box indicates the user that he/she needs to start the annotation process. The user is invited to select a trajectory in the map, where a combination of color and arrows shows the movement direction and intensity of movement features like speed or direction variation. The top right box interactively shows the numerical values of the selected features in the plot, supporting the user in the annotation process. The process is finished when the total budget of trajectories are annotated.

We detail and exemplify how the user is assisted by the ANALYTiC tool in the active learning process as follows. The ANALYTiC tool combines map visualization aspects with charts regarding the trajectory's features and the semantic labels added by the user. The map display needs to cope with some challenges. First of all, the display of all the trajectories data is, in most cases, unfeasible due to a large number of objects to be shown, and the view would not be meaningful for the user. This is the reason why we decided to show only a randomly chosen first small set of trajectories to be labeled. Therefore, the map only displays a maximum number of trajectories at the same time, and this number is limited by the bag size $\mathcal{L}$.

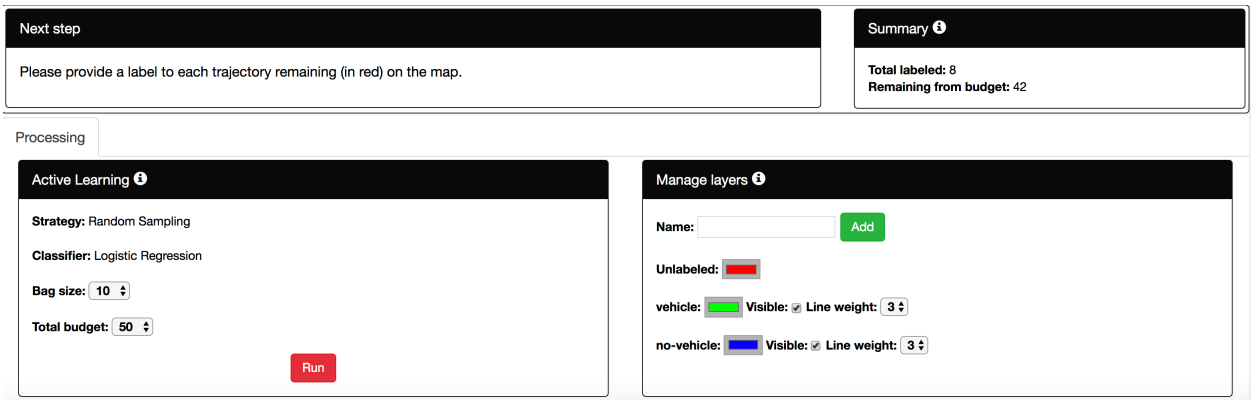
Furthermore, the map panel visualization needs to be effective in showing in a compact way the actual movement with the trajectory and point features. For this reason, we implemented two visualization solutions to ease the movement understanding of the annotator: (i) the actual movement is explicitly displayed with moving arrows that are positioned according to the movement direction; and (ii) the colors of the lines are displayed in saturation grades of red color reflecting the value of the point feature: a low value is colored with a light intensity of red while higher values are colored with a more intense red. The point feature's actual values are detailed using a line chart that shows the values following the temporal order. The tool provides an interaction between the line chart and the map: when the user clicks on a value on the line chart, the corresponding geographical position is displayed on the map by plotting a red pin on it. This feature helps the user to connect the point features observed in the plots with the geographical position of the moving object to better understand the moving object's behavior, its location, and to perform a more accurate labeling. In the case of the animals dataset, for example, the user could verify when the trajectory is geographically close to a water resource and showing a low speed behavior. In this case the expert may conclude that the trajectory belongs to a cattle and annotate the trajectory accordingly.

Another example of how the platform can assist the user in the labeling task is exemplified in Figure 6. This figure shows the elements available to the user to evaluate and interact with a trajectory chosen by the

Figure 5: Overview on ANALYTiC panels.



(a) Top panel overview.

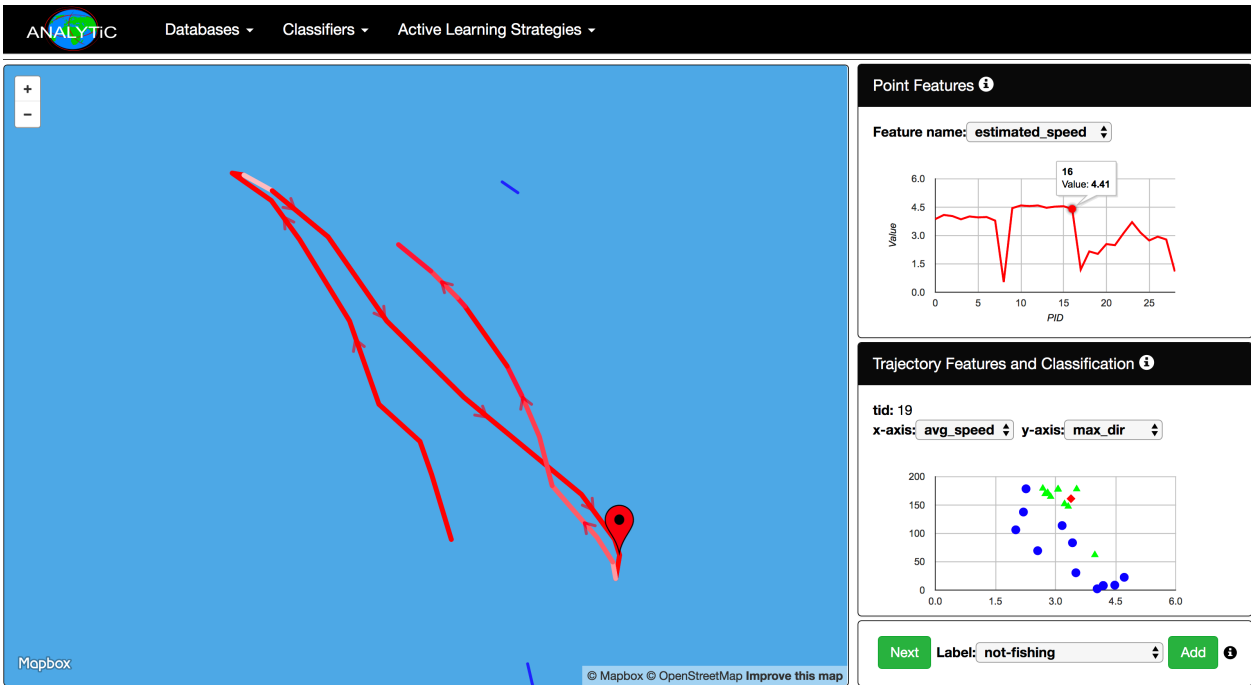


(b) Bottom panel overview.

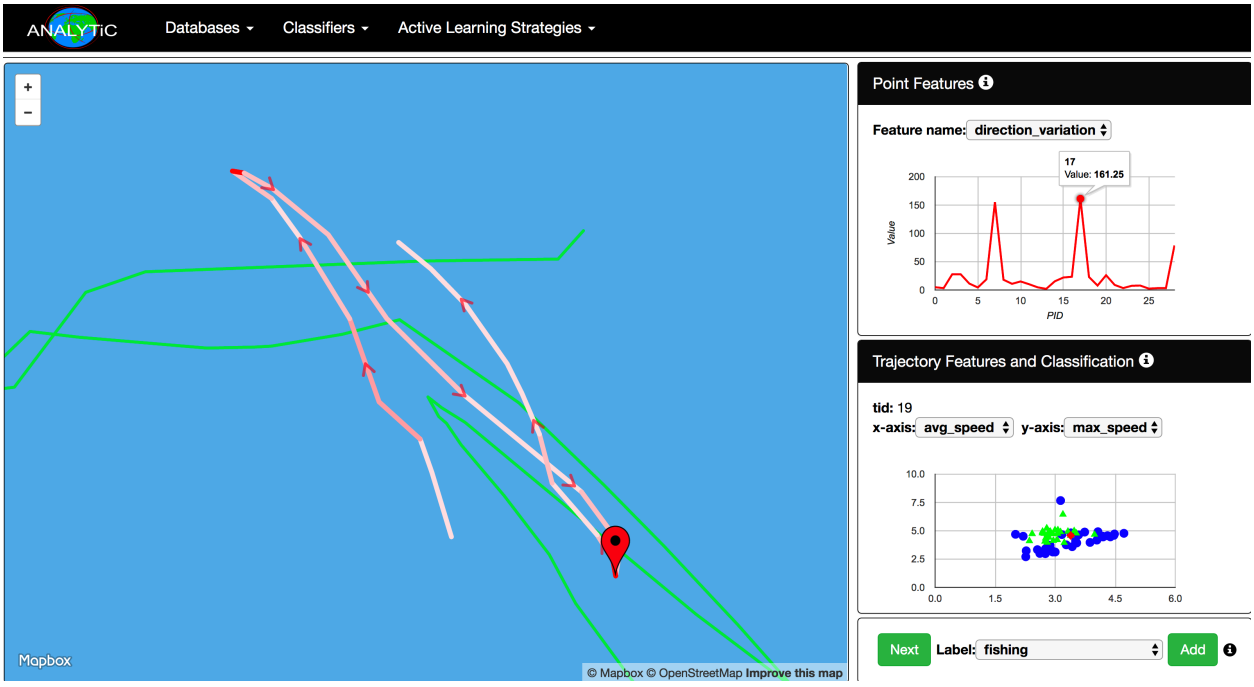
AL strategy to be labeled. Figure 6(a) shows an example of a trajectory from the fishing vessels dataset where the user visualizes the estimated speed in the sequence of points. Notice the dark red representing higher speed. The chart in the bottom part (i.e. Trajectory features and classification) is designed for the user to explore correlations between the trajectory features and the labels assigned to the trajectories. In Figure 6(a), after adding labels for some trajectories, the user may explore a plot of the average speed (e.g. avg_speed) with the maximal direction variation (e.g. max_dir) and observe that the fishing pattern occurs with more frequency when the average speed is low and the maximal direction variation is high. After labeling more trajectories using the chosen AL strategy the user can keep exploring the trajectory features space and this situation is exemplified in Figure 6(b). The figure shows the same trajectory visually represented by its direction variation. Again, the red intensity changes accordingly to the values of each point feature value. Other conclusions could be derived from Figure 6(b) regarding the trajectory features.

For example, by exploring and plotting the average speed and the maximal speed (e.g. max_speed) in the trajectory features chart, the user could realize that some sort of linear separability is observed between these features. These plots therefore support the user in getting detailed information and correlations whenever he/she needs more insights in the trajectory labeling.

Figure 6: ANALYTiC tool elements.



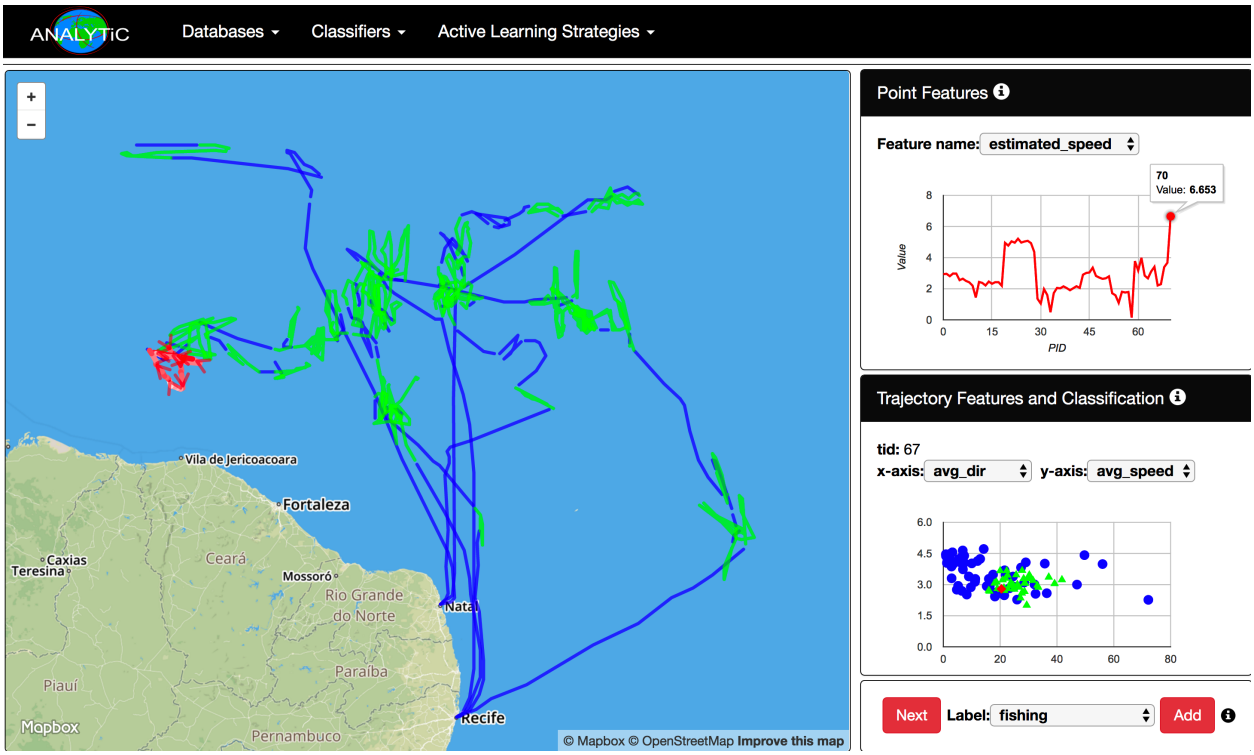
(a) Analysis of estimated speed for the fishing vessels dataset.



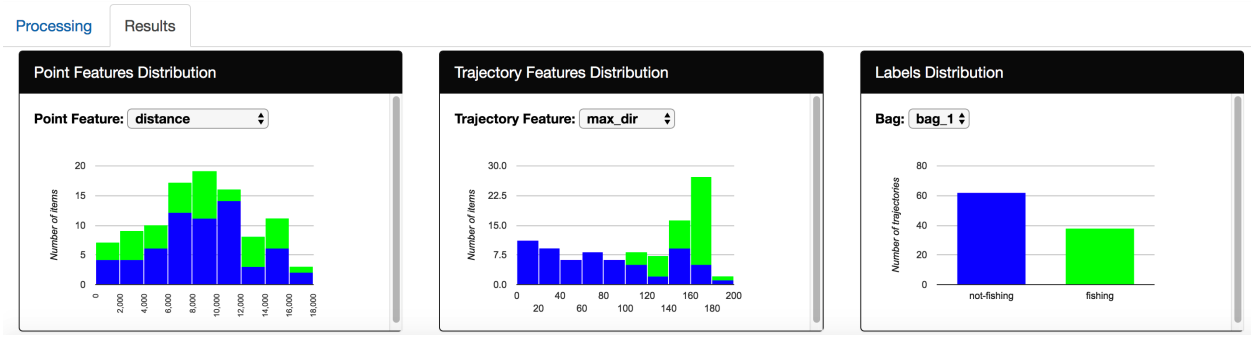
(b) Analysis of direction variation for the fishing vessels dataset.

After labeling all the trajectories required by the AL strategy, the annotator can visualize the results of the classification of all trajectories in the entire dataset. The same concept of bags, used to indicate the group of trajectories to be labeled, is used for visualizing the results, avoiding a huge overlapping of trajectories on the map. Figure 7(a) shows an example of a bag of trajectories labeled as fishing and not fishing vessels, while Figure 7(b) shows the charts with results obtained by an AL strategy. The user can explore three different charts as the result of a simulation: (i) a histogram for point features; (ii) a histogram for trajectory features; and (iii) the labels distribution. The ANALYTiC tool is available for testing at the URL `https://bigdata.cs.dal.ca/resources`.

Figure 7: ANALYTiC results display.



(a) A bag with trajectories labeled by the ANALYTiC tool.



(b) Output analysis generated by a simulation with an AL strategy and a classifier.

6 Conclusions

The problem of annotating trajectory data is well recognized in the literature, and many efforts are being devoted to this objective. However, the manual annotation of large datasets is unfeasible and therefore machine learning classification techniques can help in trying to automatically infer trajectory labels by learning from a labeled training set. However, human labeled trajectories datasets are difficult to obtain, due to the human effort needed to annotate them semantically. Is there a machine learning method that will allow an automatic trajectory classification by minimizing the number of required human labeled trajectories? Is this machine learning method effective for trajectory data? In trying to answer these questions we propose the use of active learning, a machine learning method that attains high performance with fewer labeled examples when compared to classical supervised strategies. The burden of the manual labeling process is further alleviated by the proposed ANALYTiC tool, a web-based interactive system that assists the user through the steps of the active learning process with a specific focus on the labeling trajectories task.

We showed, through a series of empirical evaluation experiments in 5 trajectories datasets, how active learning strategies choose the best subset to annotate (namely UNC and QBC) and perform significantly better than a standard non-active learning strategy based on a random (RND) choice of the subset to be labeled. We can, therefore, conclude that active learning is effective in reducing the trajectories datasets to be labeled. This result motivates the second contribution described in the paper, the ANALYTiC tool. This web-based visual interface is capable of supporting the domain expert through the active learning process and specifically in the trajectory annotation through a set of visual solutions that ease the labeling inference task.

This work is a novel direction in combining visual techniques with machine learning for movement data. We have several interesting directions to follow for further studies. The addition of segmentation algorithms as the first task of this process is certainly one step further to build a more comprehensive system. We also intend to improve the ANALYTiC system by allowing the user to add more geographical features related to the domain: this will create more meaningful point and trajectory features that can help in the classification task. Finally, an empirical analysis of multi-class trajectory databases would be conducted aiming to verify how the AL strategies perform under this condition.

References

[1] Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 1–9, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[2] Vania Bogorny, João Francisco Valiati, Sandro da Silva Camargo, Paulo Martins Engel, Bart Kuijpers, and Luis Otávio Alvares. Mining maximal generalized frequent geographic patterns with knowledge constraints. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pages 813–817, 2006.

[3] Gerben Klaas Dirk De Vries and Maarten Van Someren. Machine learning for vessel trajectories using compression, alignments and domain knowledge. *Expert Syst. Appl.*, 39(18):13426–13439, December 2012.

[4] Renato Fileto, Cleto May, Chiara Renso, Nikos Pelekis, Douglas Klein, and Yannis Theodoridis. The Baquara^2 knowledge-based framework for semantic enrichment and analysis of movement data. *Data Knowl. Eng.*, 98:104–122, 2015.

[5] J. Garcia, O. P. Concha, J. M. Molina, and G. D. Miguel. Trajectory classification based on machine-learning techniques over tracking data. In *2006 9th International Conference on Information Fusion*, pages 1–8, July 2006.

[6] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual analytics: Definition, process, and challenges. In *Information visualization*, pages 154–175. Springer Berlin Heidelberg, 2008.

[7] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 3–12, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[8] Lin Liao, Donald J. Patterson, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Artif. Intell.*, 171(5-6):311–331, April 2007.

[9] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. Technical report, Swedish Institute of Computer Science, 2009.

[10] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, Yannis Theodoridis, and Zhixian Yan. Semantic trajectories modeling and analysis. *ACM Computing Surveys*, 45(4):42:1–42:32, 2013.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] Maria E. Ramirez-Loaiza, Manali Sharma, Geet Kumar, and Mustafa Bilgic. Active learning: an empirical study of common baselines. *Data Mining and Knowledge Discovery*, pages 1–27, 2016.

[13] D. Reker, P. Schneider, and G. Schneider. Multi-objective active machine learning rapidly improves structure–activity models and reveals new protein–protein interaction inhibitors. *Chemical Science*, 7, 2016.

[14] Chiara Renso, Stefano Spaccapietra, and Esteban Zimányi, editors. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press, 2013.

[15] Salvatore Rinzivillo, Fernando de Lucca Siqueira, Lorenzo Gabrielli, Chiara Renso, and Vania Bogorny. Where have you been today? annotating trajectories with daytag. In *SSTD 2013, Germany,2013. Proceedings*, pages 467–471, 2013.

[16] Lívia Ruback, Marco Antonio Casanova, Alessandra Raffaetà, Chiara Renso, and Vânia Maria P. Vidal. Enriching mobility data with linked open data. In *Proceedings of IDEAS 2016, Canada, July 2016*, pages 173–182, 2016.

[17] Neil Rubens, Mehdi Elahi, Masashi Sugiyama, and Dain Kaplan. Active learning in recommender systems. In *Recommender Systems Handbook*, pages 809–846. 2015.

[18] Claude Sammut and Geoffrey I. Webb. *Encyclopedia of Machine Learning*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[19] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[20] James J. Thomas and Kristin A. Cook. A visual analytics agenda. *IEEE Comput. Graph. Appl.*, 26(1):10–13, January 2006.

[21] Xinchang Zhang, Guowei Luo, Guangjing He, and Liyan Chen. A multi-scale residential areas matching method using relevance vector machine and active learning. *ISPRS International Journal of Geo-Information*, 6(3), 2017.

[22] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. Understanding mobility based on gps data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pages 312–321, New York, NY, USA, 2008. ACM.

[23] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. Understanding mobility based on GPS data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pages 312–321, New York, NY, USA, 2008. ACM.

[24] Yu Zheng, Like Liu, Longhao Wang, and Xing Xie. Learning transportation mode from raw gps data for geographic applications on the web. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 247–256, New York, NY, USA, 2008. ACM.