Investigation of initialization strategies for the Multiple Instance Adaptive Cosine Estimator

James Bocinsky, Connor McCurley, Daniel Shats, Alina Zare

SPIE.

Investigation of Initialization Strategies for the Multiple Instance Adaptive Cosine Estimator

James Bocinsky, Connor McCurley, Daniel Shats, and Alina Zare

Department of Electrical & Computer Engineering, University of Florida

ABSTRACT

Sensors which use electromagnetic induction (EMI) to excite a response in conducting bodies have long been investigated for subsurface explosive hazard detection. In particular, EMI sensors have been used to discriminate between different types of objects, and to detect objects with low metal content. One successful, previously investigated approach is the Multiple Instance Adaptive Cosine Estimator (MI-ACE). In this paper, a number of new initialization techniques for MI-ACE are proposed and evaluated using their respective performance and speed. The cross validated learned signatures, as well as learned background statistics, are used with Adaptive Cosine Estimator (ACE) to generate confidence maps, which are clustered into alarms. Alarms are scored against a ground truth and the initialization approaches are compared.

Keywords: Multiple instance learning, explosive hazard detection, adaptive cosine estimator, electromagnetic induction, bagging, K-Means, Gaussian Mixture Model, ROC curve

1. INTRODUCTION

Many machine learning approaches use individual samples with their corresponding classification label to learn a representative of each class for classification. For some applications, such as explosive hazard detection[1234] or drug activity prediction,[5] this is unfeasible due to the ambiguity of the data. The data does not present itself in a way that can be labeled at the sample level. A machine learning framework known as multiple instance learning (MIL) was formalized by Dietterich et al. to handle applications with ambiguously labeled data.[5] In this framework, instance level labels are not available, so the data is grouped into bags, with bag level labels. Although there are other variations to MIL, the original MIL framework assumes that a "positive" bag corresponds to a bag that has at least one instance corresponding to a target class of interest. A "negative" bag corresponds to a bag that contains only instances that belong to the background, non-target class. A typical process of using MIL is to determine the instances and their salient features from the positive bags that help differentiate the detection of unknown instances.

In the context of explosive hazard detection, one reason MIL is often used is because the area of the explosive hazard's EMI response is unknown. It is true that the location of a target is known, but it is unknown how the magnitude of the response will reduce over space, or how much spatial area the target response can be detected. This invokes one reason why MIL fits this application; it is simple to label a spatial region as containing a response from a explosive hazard as a "positive" bag, and other regions where no explosive hazards exist as "negative" bags. Labeling

---

Further author information: (Send correspondence to James Bocinsky and Alina Zare)
James Bocinsky: E-mail: jbocinsky1@gmail.com
Alina Zare: E-mail: azare@ece.ufl.edu

individual responses, known as instances in MIL, along a physical sweep of a handheld sensor is nearly impossible. Additionally, another benefit of using MIL for explosive hazard detection is to allow the algorithm to learn from the data what is a good representative of an explosive hazard. Laboratory measurements can be measured or physics based models like the Discrete Spectrum of Relaxation Frequencies[6] can be generated to model an explosive hazard response. However, it is often seen that the response of the electromagnetic induction (EMI) sensors used for explosive hazard detection act differently when an explosive hazard is buried than in the lab or physics based models. So although these models may be accurate in the lab or in ideal physical conditions, various soil responses often make these models inaccurate to real world responses. More so, MIL can learn a representative that is normalized to the soil and model the local soil properties. This way, the MIL algorithm can learn a representative that maximizes detection in the soil being investigated.

The Multiple Instance Adaptive Cosine Estimator (MI-ACE) algorithm is a MIL algorithm that makes use of these properties, and even more so focuses on maximizing target detection using the Adaptive Cosine Estimator (ACE) detection statistic. This means that the algorithm does not only determine a signature that is most like a target signature and unlike the background data, but as well learns a target signature that maximizes the ACE detection statistic. The ACE statistic has nice properties for explosive hazard detection. ACE applies a transformation to every test sample with respect to the background data, known as data whitening.[7][8][9] This is done by subtracting the background mean and multiplying by the inverse covariance matrix of the background. This results in a zero mean and unit variance data point with respect to the background. ACE also normalizes the similarity by the magnitude of both the target signature and the test sample. This means that regardless of magnitude, the relative feature vector values or shape of the test sample is responsible for the similarity measure. This is highly beneficial for detecting low metal explosive hazards where the magnitude response may be low. Furthermore, explosive hazards buried at different depths often have similar response shape, but at different magnitudes. Since the shape of the response is used for detection, these explosive hazards can still be found. With these properties, it can be seen that the MI-ACE algorithm has many desired aspects of an ideal algorithm for explosive hazard detection.

One aspect of MI-ACE that is crucial to the performance of the algorithm is the initialization procedure. The optimization part of the algorithm is dependent on there being a reasonably well initialized target signature. If not, the algorithm may converge to a poor representative. The principle initialization technique outlined in the original MI-ACE publication[10] is to initialize the instance from the positive bag that maximizes the MI-ACE objective function. This requires searching through all of the positive instances, finding each positive instance's positive bag representative, computing the ACE similarity to each of the background samples, and finally computing the objective function. This is computationally expensive with a complexity of $O((N^+)^2 + N^+N^-)$, where N^+ and N^- are the total number of positive and negative instances respectively.

In this paper, three new initialization techniques are investigated to improve performance and reduce the computational cost of the initialization process. Clustering techniques are investigated to reduce the number of samples that must be searched for initialization. As well, a new statistic, the multiple instance cluster rank is proposed to reduce the computation complexity of initializing a target representative. The performance of these techniques with the initialized and optimized signatures, for three data sets, and two sensors are shown along with a computational cost analysis.

2. METHODS

2.1 Multiple Instance Adaptive Cosine Estimator

The Multiple Instance Adaptive Cosine Estimator (MI-ACE) was originally proposed to solve common problems of spatial inaccuracy in training data in target detection applications.[10] In order to use a detection metric like ACE, a target signature must be known prior to performing detection. Techniques to estimate target representatives can be measured in a laboratory setting, but are often unrealistic and not representative of a target in various conditions and environments. Alternatively, a target representation may be extracted directly from the data itself. Often times when this is done, the extracted representation does not contain meaningful features to differentiate it from the background and may not provide the desired performance. Lastly, it is often times difficult or even impossible to extract a target representation from a dataset. For the explosive hazard detection problem, the boundaries of an explosive hazard's response within a physical sweep of the sensor are challenging to obtain, and thus determining where to extract a target representation is nonviable. The MI-ACE algorithm addresses these problems and is able to learn a target signature that is optimal for the ACE detection metric.

2.1.1 Method

The MI-ACE algorithm[10] follows the multiple instance learning framework where the labels of the data are at the bag level. With this, let $\mathbf{X} = [\mathbf{x_1}, ..., \mathbf{x_N}]$ be training data with each sample, $\mathbf{x_i}$ being a vector with dimensionality d. The data is grouped into K bags $\mathbf{B} = \{\mathbf{B_1}, ..., \mathbf{B_K}\}$ with labels, $L = \{L_1, ..., L_K\}$, where $L_j \in \{0, 1\}$. A bag is considered positive, $\mathbf{B_j^+}$, with label, $L_j = 1$, when there exists at least one instance in bag j that is from the target class. Additionally, a bag is considered negative, $\mathbf{B_j^-}$, with label $L_j = 0$, if all instances in bag j are from the background class. The number of instances in both positive and negative bags does not need to be fixed.

With this formulation, the goal of MI-ACE is to estimate a target signature, $\mathbf{s}$, that maximizes the detection statistic of the target instances in the positive bags while minimizing the detection statistic of all negative instances. This is accomplished by maximizing the objective shown in eq. (1),

$$\arg\max_{\mathbf{s}} \frac{1}{N^+} \sum_{j:L_j=1} D_{ACE}(\mathbf{x_j^*}, \mathbf{s}) - \frac{1}{N^-} \sum_{j:L_j=0} \frac{1}{N_j^-} \sum_{\mathbf{x_i} \in B_j^-} D_{ACE}(\mathbf{x_i}, \mathbf{s}), \tag{1}$$

where N^+ is the number of positive bags, N^- is the number of negative bags, and N_j^- is the number of instances in negative bag j. $\mathbf{x_j^*}$ is the positive instance selected from bag j that is most like the target signature, $\mathbf{s}$, known as the bag representative,

$$\mathbf{x_j^*} = \arg\max_{\mathbf{x_i} \in \mathbf{B_j^+}} D(\mathbf{x_i}, \mathbf{s}). \tag{2}$$

The detection statistic, ACE, shown in eq. (3), is the projection of a test sample, $\mathbf{x}$, onto a known target signature, $\mathbf{s}$, in a whitened coordinate space. Again, the whitening is done using the background covariance, $\mathbf{\Sigma_b^{-1}}$, and background mean, $\boldsymbol{\mu_b}$, to transform the data to have zero mean and a uniform, unit variance with respect to the background. ACE is normalized by not only the target signature, $\mathbf{s}$, but also the whitened test sample, $\mathbf{x}$, as well. With this, the magnitude of the

test sample will not affect the statistic, and only the shape of the feature vector contributes to the statistic.

$$D_{ACE}(\mathbf{x}, \mathbf{s}) = \frac{\mathbf{s^T \Sigma_b^{-1}(x - \mu_b)}}{\sqrt{\mathbf{s^T \Sigma_b^{-1} s}}\sqrt{\mathbf{(x - \mu_b)^T \Sigma_b^{-1}(x - \mu_b)}}} \tag{3}$$

To estimate the target signature, the objective function in eq. (1) is maximized. To accomplish this, the algorithm is broken up in to two primary steps, initializing a target signature, and then optimizing that signature using a single instance from each positive bag, also known as the bag representative, $\mathbf{x_j^*}$. The original initialization method computes the objective function for all of the positive instances and whichever instance provides the largest objective function becomes the initialized target signature. Although this instance may provide the highest objective function value, it may not be optimal for all of the positive instances within the data. So considering this, optimization is done using the update equation shown in eq. (4). Here $\hat{\mathbf{s}}$, $\hat{\mathbf{x}}_\mathbf{j}^*$, and $\hat{\mathbf{x}}_\mathbf{i}$ are the whitened signature, whitened bag representative, and whitened negative instance respectively.

$$\hat{\mathbf{s}} = \frac{\mathbf{t}}{||\mathbf{t}||} \quad \text{where} \quad \mathbf{t} = \frac{1}{N^+} \sum_{j:L_j=1} \hat{\mathbf{x}}_\mathbf{j}^* - \frac{1}{N^-} \sum_{j:L_j=0} \frac{1}{N_j^-} \sum_{\mathbf{x_i} \in B_j^-} \hat{\mathbf{x}}_\mathbf{i} \tag{4}$$

To optimize the initialized signature, the signature, $\hat{\mathbf{s}}$, is iteratively updated using eq. (4). In each iteration, the current bag representatives, $\mathbf{x_j^*}$, are determined given the current estimated target signature. The bag representatives are averaged and then the average of the background samples is subtracted away. The average background will not change from iteration to iteration so this term can be precomputed. Finally the target signature is normalized and the updated target signature has been computed.

2.2 Alternative Initialization Approaches

Additional initialization approaches using clustering methods have been investigated to determine if either performance or run time can be improved for MI-ACE. The original initialization approach is to initialize the instance from a positive bag that maximizes the objective function. This requires the MI-ACE algorithm to search through all of the positive instances and compute the objective function. This is a computational expensive process, $O(N^+(N^+ + N^-))$, where N^+ and N^- are the total number of positive and negative instances respectively. Alternative initialization approaches using clustering are explored to reduce computation time by using the cluster centers as target candidates instead of every positive instance. With this, the algorithm does not need to search through as many candidate points to initialize a target signature. Additionally, the initialization technique will learn a representation of the target signature that is representative of a subspace of the data, instead of initializing a single instance that may or may not represent a greater region of the target class.

2.2.1 K-Means

The first approach, referred to as K-Means, uses the K-Means clustering algorithm[11] to group all of the data, regardless of bag structure, into K clusters. Then, this initialization approach picks the cluster center that maximizes the MI-ACE objective function as the initialized target signature.

This computation complexity is $O(Ki(N^+ + N^-) + K(N^+ + N^-))$, where K is the number of clusters, N^+ and N^- are the number of positive and negative instances, respectively, and i is the number of iterations until K-Means converges. The first term corresponds to K-Means clustering, and the second term corresponds to determining the cluster centers that maximize the objective function. As long as the number of clusters, K, and the number of iterations i, remains small, the K-Means approach will have less of a computational cost than the original initialization method. This way the algorithm only needs to search through K candidates instead of N^+ candidates to initialize a target signature.

2.2.2 Ranked K-Means

The second approach, referred to as Ranked K-Means, uses the K-Means clustering algorithm[11] to create K clusters, regardless of bag structure. Instead of using the original objective function to score the cluster centers, a new multiple instance cluster rank is proposed to further reduce the computational cost. The multiple instance cluster rank of the kth cluster, is the sum of the proportions of the elements in cluster k. The three terms of the rank are the proportion of positive bags that have an instance in cluster k, the proportion of instances in cluster k that came from a positive bag, and the proportion of instances in cluster k that came from a negative bag. This is formally defined below in (5)

$$Rank_{MIC}(k) = \frac{w^{B^+}\left(\frac{N_k^{B^+}}{N^{B^+}}\right) + w^+\left(\frac{N_k^+}{N^+}\right) - w^-\left(\frac{N_k^-}{N^-}\right) + w^-}{w^{B^+} + w^+ + w^-}, \qquad (5)$$

where N^{B^+}, N^+, N^- are the total number of positive bags, positive instances, and negative instances, respectively. $N_k^{B^+}$, N_k^+, N_k^- are the number of positive bags that have at least one instance in cluster k, the number of positive instances in cluster k, and the number of negative instances in cluster k, respectively. Finally, the weights w^{B^+}, w^+, and w^- are positive hyperparameter weights that are set based on the distribution of instances in the constructed positive bags. If it is believed that the positive bags contain a majority of positive instances, then the first two weights should be higher than the last weight. Furthermore, if the positive bags contain a majority of negative instances, the last weight should dominate to require a minimal number of negative instances belonging to the cluster center that is initialized. The equation adds w^- and is divided by the sum of the weights to force the rank to be from $[0, 1]$.

The computation complexity for this technique is $O(Ki(N^+ + N^-) + K)$, where K is the number of clusters, N^+ and N^- are the number of positive and negative instances, respectively, and i is the number of iterations until K-Means converges. In this initialization technique, the first term, corresponding to K-Means, dominates the complexity. The second term corresponds to determining the cluster center that maximizes the multiple instance cluster rank. Since all of the data proportions to compute the rank come straight from the clustering results, no additional computation complexity is needed to determine the cluster rank for each cluster. This is an indexed matrix lookup and therefore constant time, and thus not included in the second term of the computation complexity.

2.2.3 Multiple Instance Cluster Regression (MI-ClusterRegress)

The MI-ClusterRegress algorithm[12] clusters all of the data regardless of bag structure into K clusters using a Gaussian Mixture Model (GMM).[13] Then, an exemplar point is created in each positive bag for each of the K distributions. An exemplar point is a weighted average of the instances

in a bag, where the weights correspond to the membership of that instance to the corresponding cluster. Namely, the exemplar point for cluster k within bag j, denoted as $\hat{\mathbf{B}}_{\mathbf{j}}^{\mathbf{k}}$, is the average of all data points, $\mathbf{x_i}$, in bag j weighted by their memberships in cluster k, denoted by R_i. Then, using the exemplars, a regression model is fit to each cluster k, using the exemplar points from each bag that correspond to cluster k. The eq.s (6) - (8) demonstrate how to compute the membership relevance, R_i, for each instance $\mathbf{x_i}$. In the original algorithm, these exemplar points are then used to train K separate regression models to allow each distribution to have their own local regression model.

$$r_i = \mathrm{P}(\mathbf{x_i} \in c_k | \mathbf{B_j}; \boldsymbol{\theta_{c_k}}), \forall i \tag{6}$$

$$z := \sum_{i=1}^{N_j^+} r_i \tag{7}$$

$$R_i := \frac{r_i}{z}, \forall i \tag{8}$$

Here, $\mathbf{x_i} \in c_k$ means that instance $\mathbf{x_i}$ was generated by cluster c_k. This probability in equation (6) can be computed using the learned parameters, $\boldsymbol{\theta_k}$, from each of the Gaussian Mixture Model distributions. Then a normalization term z is computed for the bag, $\mathbf{B_j}$, as the sum of all memberships from bag, $\mathbf{B_j}$. Here, N_j^+ is the number of instances in bag j. Lastly, each instance's membership is normalized by z to form the relevance, R_i, of an instance belonging to the kth distribution in bag j.

This algorithm has been incorporated into this initialization method, referred to as MI-CR. In this initialization method, the clustering portion of MI-ClusterRegress, as well as creating the exemplar points, is used to reduce the number of instances that must be searched to initialize a target concept. Additionally, the created exemplar points are a combination of the instances in a positive bag, so the initialized exemplar point may be better at representing the variations in target representatives rather than using a single instance as the initialized target concept.

The computation complexity for MI-CR is $O(Ki(N^+ + N^-) + KN^{B^+}(N^+ + N^-))$, where K is the number of clusters, N^+ and N^- are the number of positive and negative instances, respectively, N^{B^+} is the number of positive bags, and i is the number of iterations until Expectation Maximization for GMM converges. The first term corresponds to GMM's complexity. The second term corresponds to how many exemplar points are being considered as a potential target signature. For each bag, there are K exemplar points generated, so a total of KN^{B^+} exemplar points are generated. Then the objective value is computed with complexity of $(N^+ + N^-)$ for each of the exemplar points. This will dominate the computation time of MI-CR if there are many bags or many instances, but the computation time can also be largely affected by the stopping condition threshold used for the GMM.

MI-ACE Initialization Methods Time Complexity	
Method	Time Complexity
Original	$O(N^+(N^+ + N^-))$
K-Means	$O(Ki(N^+ + N^-) + K(N^+ + N^-))$
Ranked K-Means	$O(Ki(N^+ + N^-) + K)$
MI-CR	$O(Ki(N^+ + N^-) + KN^{B^+}(N^+ + N^-))$

3. EXPERIMENTAL RESULTS

A data collection using handheld electromagnetic induction (EMI) sensors was used to test the various initialization approaches. This data collection consisted of various explosive hazards, and for testing purposes was broken into three groups. The three groups were all high metal targets, all low metal targets, and all targets including no metal. Additionally, two different EMI sensors were used, sensor A and sensor B. Experiments were run with both the initialized signature and the optimized signature. With this configuration, a total of 12 experiments were run, six for the initialized signature, and six with the optimized signature.

Number of Targets in Data Subsets	
Data Subset	Number of Targets
Sensor A (Metal)	39
Sensor A (Low)	70
Sensor A (all)	167
Sensor B (Metal)	9
Sensor B (Low)	20
Sensor B (all)	39

The algorithm was trained using lane based cross validation. For example, if a site consisted of five lanes, during one fold of cross validation, four lanes would be used for training and the other lane would be used for test. Each lane consisted of an unknown multiple of grids, where each grid contained a single explosive hazard. Each positive bag was generated from a single grid. The samples inside a predicted response radius of the target's spatial center were taken to be the samples in the positive bag. A single negative bag was generated using all samples from the low and no metal grids blank sweeps. The blank sweeps were created at the beginning portion of a grid. It was seen that high metal grids would often have target response bleed over in to the blank sweeps, so only the low and no metal blank sweeps were used for the negative bag.

For, only the initialized signatures were used to generate confidence maps on the test data. Namely, no optimization was done after the learned target signature was initialized. Then, optimization was included with each of the techniques. For all of the proposed techniques, five clusters were used. Once a signature was learned, the ACE similarity statistic was used to generate confidence maps, a confidence for every sample along a test sweep. The generated confidences, along with the corresponding spatial coordinates, were then passed into a mean shift clustering algorithm[14] to generate alarms. Mean shift clustering was used to group the spatial regions of the sweep that had similar confidences into separate clusters. Each cluster's center was used as the cluster's alarm location.

In practice, a larger uniform response is desired for detecting explosive hazards. The operator would be able to determine if an explosive hazard exists earlier, and be able to determine its' shape and location with more accuracy. This was taken into account for setting the score of a generated alarm. The samples that belong to a generated alarm are those in the initial mean shift cluster and those within an allowable distance, d_{halo}, of the cluster center, c_{center}. The allowable distance was set to 0.25 meters for these experiments. The score of an alarm was computed as the weighted average of the confidences that belong to that alarm, shown in equation (9).

$$Score = \frac{1}{N_c} \sum_{i=1}^{N_c} (w_i * conf_i) \tag{9}$$

Where N_c is the number of samples in the alarm, and $conf_i$ is the confidence associated with the ith sample, x_i. The weight, w_i, corresponding to the ith sample, is the spatial Euclidean distance of the sample to the center of the cluster, c_{center}, shown in equation (10). The weight is divided by the maximum allowable distance, d_{halo}, to normalize the weight from 0 to 1.

$$w_i = \frac{||x_i - c_{center}||}{d_{halo}} \tag{10}$$

This was done to boost the score of alarms that had a larger surface area, as this is desired in practice. The label of the alarm was determined by whether the cluster center fell into the expected response radius of an explosive hazard. Finally, these alarms and labels were used to generate ROC curves for the different experiments.
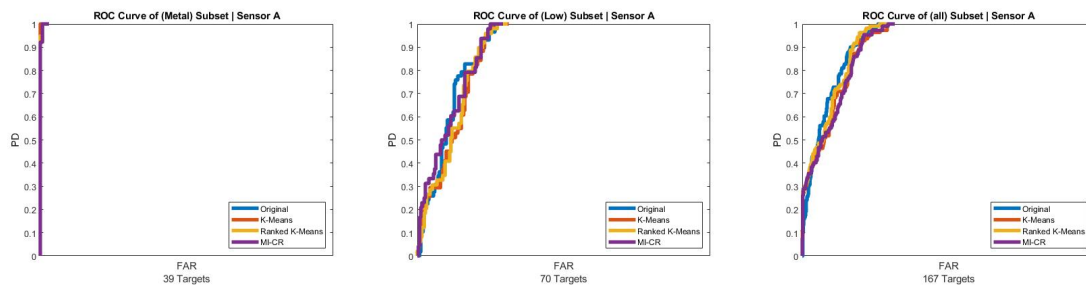

Figure 1: MI-ACE ROC curves using the initialized signature from the original initialization method and the three proposed initialization methods using Sensor A for high metal, low metal, and all explosive hazards, from left to right.
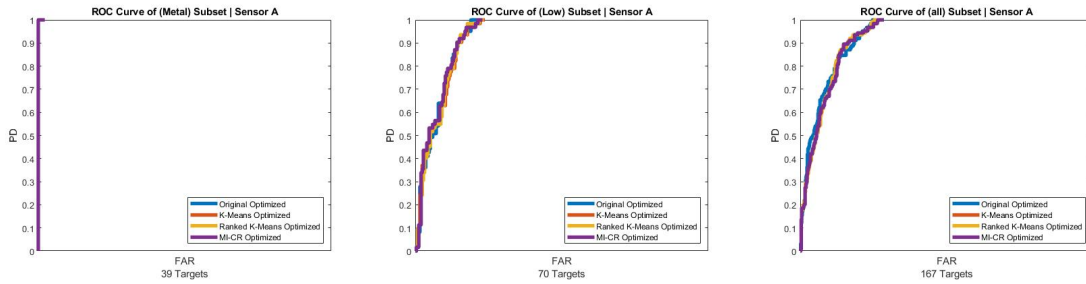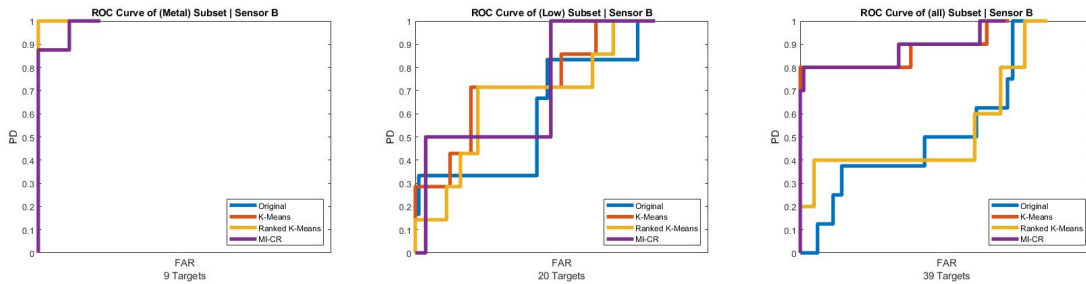
Figure 2: MI-ACE ROC curves using the corresponding optimized signature from the original initialization method and the three proposed initialization methods using Sensor A for high metal, low metal, and all explosive hazards, from left to right.
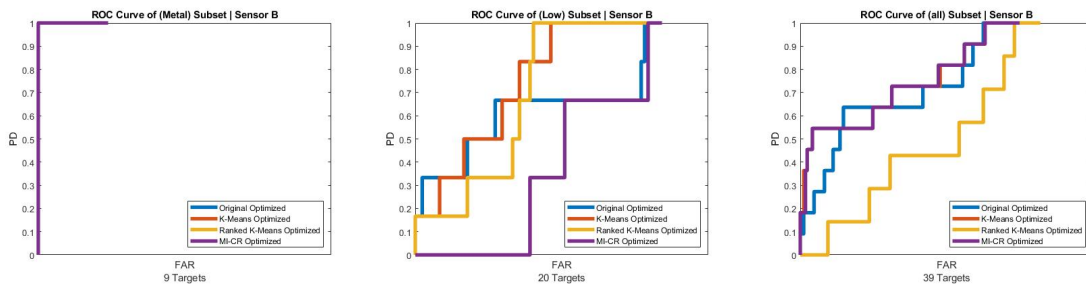

Figure 3: MI-ACE ROC curves using the initialized signature from the original initialization method and the three proposed initialization methods using Sensor A for high metal, low metal, and all explosive hazards, from left to right.

Figure 4: MI-ACE ROC curves using the corresponding optimized signature from the original initialization method and the three proposed initialization methods using Sensor B for high metal, low metal, and all explosive hazards, from left to right.

Initialization Run Time Comparison [ms]				
Experiment	Original	K-Means	Ranked K-Means	MI-CR
Sensor A (Metal)	816.0	6.9	10.5	37.1
Sensor A (Low)	2,216.7	28.6	18.2	98.3
Sensor A (all)	40,431.4	65.1	60.0	250.0
Sensor B (Metal)	42.2	1.0	1.4	9.4
Sensor B (Low)	211.5	1.7	4.7	81.7
Sensor B (all)	678.5	15.4	7.4	64.6
Average	**7,399.4**	**19.8**	**17.0**	**90.2**

4. DISCUSSION AND FUTURE WORK

The focus of this paper is to analyze the various initialization approaches proposed. In most cases the ROC curve generation process works as expected. It can be seen in figure 5 that an alarm is generated in the center of the high confidence region as expected. The alarm generation is not perfect though, and has it's flaws. The problem with the alarm generation is twofold, first the mean shift clustering does not always perform as expected. For example taking the downtrack confidence map and generating it's alarms, shown in figure 6, we can see that mean shift generates multiple alarms. It appears clear that there should only be one alarm, but mean shift generates an additional alarm. Second, the response radius of an explosive hazard is unknown and changes based on preprocessing and the algorithm being used. So a generated alarm that should be truly considered a true target alarm, can fall just outside what was determined to be the ground truth and cause a false alarm. An example of this is shown in figure 7, where the alarm generated came from the response of the explosive hazard, but due to an unknown true response radius, the alarm is considered a false alarm.

With these examples in place, this promotes a need for creating a variable ground truth radius. The radius could be based on explosive hazard type or even algorithm, as it has been noticed that the response radius changes with different algorithms. If the ground truth can be variable, the ROC generation curves would be more accurate, and a more definitive analysis can be done.
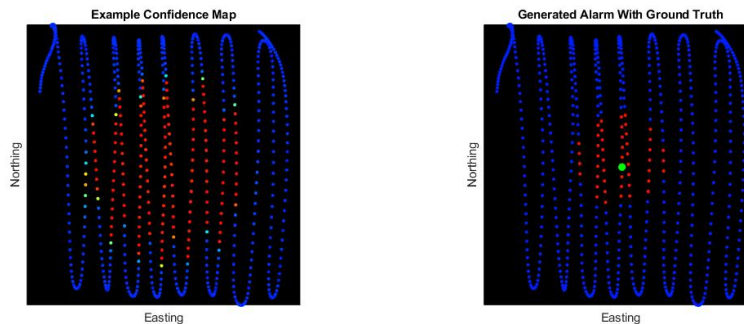

Figure 5: An example confidence map (left) with it's corresponding alarm generation plotted over the ground truth (right). The Alarm generation works as expected.
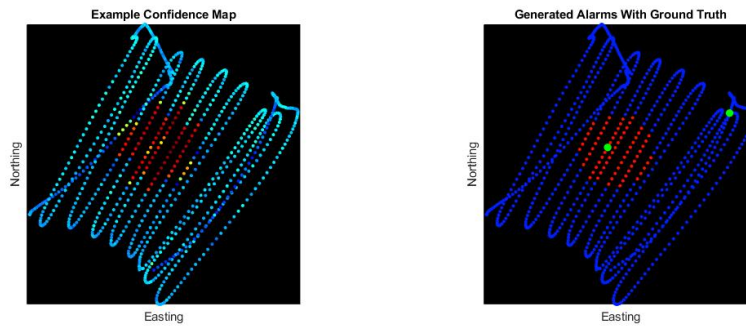
Figure 6: An example confidence map (left) with it's corresponding alarm generation plotted over the ground truth (right). Two Alarms are generated even though it appears clear there should be only one generated.
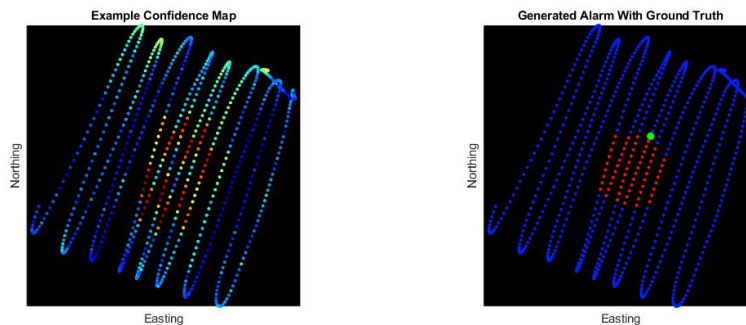

Figure 7: An example confidence map (left) with it's corresponding alarm generation plotted over the ground truth (right). The generated alarm falls just outside the ground truth and is considered as a false alarm.

With this said, the MI-ACE initialization performances, with and without optimization are compared. It can be seen that the different initialization approaches perform similarly for the majority of the experiments. All of the approaches are able to detect the high metal targets, and when the learned signature is optimized, all of the techniques find the high metal targets with no false alarms. In the experiments of sensor B, there are folds of the cross validation where the testing set has target types that do not exist in the training set. This is why it is believed that the initialized signature ROC with the all subset of sensor B, shows K-Means and MI-CR performing better than the original and Ranked K-Means. It is believed that K-Means and MI-CR are able to initialize a signature that generalizes the target class better. Whereas the original initialization technique chooses a single sample that maximizes the objective function for the training set data, which may not generalize well to the target class as a whole or the target types in the test set. Additionally, it

is expected that when there are more targets available for training, that the initialization techniques would perform more similarly like they do with sensor A which has approximately four times as many training examples as sensor B.

Furthermore, it can be seen that if the training and test data contain the same target types, the case of sensor A, any of these initialization methods will provide marginally no difference in performance when optimization is performed. This means that as long as optimization is done, the original costly initialization can be replaced with one of these variations to save run time. This is largely the case because of the nature of optimization. If two initialized signatures are similar, the optimization will likely optimize the signatures to be even more similar. This is because of how signatures are optimized. The optimized signature is the average of the positive bag representations, minus the average of the background. So often times, different initialized signatures will select the same or similar positive bag representatives and the updated signature will start averaging towards the same result, causing the resulting optimized signatures to be very similar in nature. This was noticed by analyzing how signatures change in optimization and can be confirmed by comparing Figures 1 and 2. It can be seen that the difference in ROC curves is smaller for the optimized signatures than the initialized signatures. This is likely because the optimized signatures are more similar than the initialized signatures.

The run time analysis shows that all of the proposed techniques are faster than the original, and in the case of the K-Means clustering techniques, on average they are faster on the order of 100. Furthermore, this is consistent with the time complexity analysis done in section 2.2 and shows that with a decrease in run time, the same performance can be obtained with any of the proposed initialization techniques.

5. ACKNOWLEDGMENTS

REFERENCES

[1] Zare, A., Cook, M., Alvey, B., and Ho, D. K., "Multiple instance dictionary learning for subsurface object detection using handheld emi," in [*Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*], **9454**, 94540G, International Society for Optics and Photonics (2015).

[2] Karem, A. and Frigui, H., "Fuzzy clustering of multiple instance data," in [*Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*], 1–7, IEEE (2015).

[3] Alvey, B., Ho, D. K., and Zare, A., "Fourier features for explosive hazard detection using a wideband electromagnetic induction sensor," in [*Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXII*], **10182**, 101820E, International Society for Optics and Photonics (2017).

[4] Cook, M., Zare, A., and Ho, D. K., "Buried object detection using handheld wemi with task-driven extended functions of multiple instances," in [*Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXI*], **9823**, 98230A, International Society for Optics and Photonics (2016).

[5] Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T., "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence* **89**(1-2), 31–71 (1997).

[6] Wei, M.-H., Scott, W. R., and McClellan, J. H., "Landmine detection using the discrete spectrum of relaxation frequencies," in [*2011 IEEE International Geoscience and Remote Sensing Symposium*], 834–837, IEEE (2011).

[7] Mayer, R., Bucholtz, F., and Scribner, D., "Object detection by using" whitening/dewhitening" to transform target signatures in multitemporal hyperspectral and multispectral imagery," *IEEE transactions on geoscience and remote sensing* **41**(5), 1136–1142 (2003).

[8] Kraut, S., Scharf, L. L., and McWhorter, L. T., "Adaptive subspace detectors," *IEEE Transactions on signal processing* **49**(1), 1–16 (2001).

[9] Wu, J.-C. and Wu, K.-B., "Two-stage process for improving the performance of hyperspectral target detection," in [*2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*], 1–4, IEEE (2016).

[10] Zare, A., Jiao, C., and Glenn, T., "Discriminative multiple instance hyperspectral target characterization," *IEEE transactions on pattern analysis and machine intelligence* **40**(10), 2342–2354 (2018).

[11] MacQueen, J. et al., "Some methods for classification and analysis of multivariate observations," in [*Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*], **1**(14), 281–297, Oakland, CA, USA (1967).

[12] Wagstaff, K. L., Lane, T., and Roper, A., "Multiple-instance regression with structured data," in [*Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*], 291–300, IEEE (2008).

[13] Murphy, K. P., [*Machine learning: a probabilistic perspective*], Cram101 (2013).

[14] Fukunaga, K. and Hostetler, L., "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory* **21**, 32–40 (January 1975).