2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Choosing Classification Thresholds for Mobile Robot Coverage

Parikshit Maini and Volkan Isler
Department of Computer Science
University of Minnesota - Twin Cities
United States
{pmaini,isler}@umn.edu

***Abstract*— Many robotic coverage applications involve detection of spatially distributed targets, followed by path planning to visit them for service. In these applications, the performance of the detection algorithm can have profound effect on planning decisions and costs. Range of operation, in both space and time, for robots is typically finite over a single mission and is a common constraint that needs to be accounted for in decision making. Misclassification may result in wastage of resources and can even jeopardize the completion of a mission if the length of a path extends beyond the range of the robot.**

In this work, we develop techniques on the computation of planning-aware classification thresholds. We discuss two versions that compute binary classification thresholds as a function of planning budget and detection accuracy. We present an implementation of our methods in path planning applications for an autonomous mower and show results on real and simulated data. Our method allows upto 25% improvement in coverage as compared to standard thresholding methods.

I. Introduction

With recent advances in robotics, using mobile robots for coverage tasks has become practical. In this work, we focus on applications in which a robot must visit a set of targets to service them. In some applications, such as delivery, the locations of the targets may be known in advance. In many other cases, the targets must be detected before they can be visited. For example, in a disaster response scenario, locations of victims can be identified from aerial images. These locations may then be visited by a robot to help the victims. Similar scenarios also occur frequently in environmental, agricultural and infrastructure monitoring as well as surveillance applications and follow a two-step procedure: interpretation of sensor data to detect targets followed by coverage planning to visit them. Traditionally these two steps (detection and coverage) have been treated independently. Clearly, the detection performance has a direct impact on coverage. If the detection algorithm has too many false positives, coverage may take too long or become infeasible. On the other hand, a conservative detection algorithm with too many false negatives would result in missing targets and associated undesirable consequences. In this paper, we investigate the relationship between detection and coverage performances. Specifically, we treat detection as a binary classification problem and look for systematic ways to choose *planning-aware classification thresholds* for coverage applications.

We show that performance metrics commonly used in image classification and object recognition tasks (e.g. minimizing misclassification) may not be the best choice for coverage. Under the assumption that the targets may be located uniformly at random, we show that classification thresholds can be chosen analytically to optimize coverage performance. Specifically, we study the problem of computing a classification threshold as a function of the available planning budget. We also study the dual problem to compute a threshold that ensures a given fraction of targets is covered. We apply our methods in a weed-mowing application using an autonomous mower on agricultural fields. Our methods result in upto 25% improvement in coverage performance over standard methods that do not consider planning costs in threshold computation.

II. Related Work

Path planning based on classifier learnt information is a popular technique for mobile robot navigation in many applications. Online path planning within terrains, for instance, commonly uses visual data in addition to LIDAR and thermal imaging ([1] and the references within). The work of Huang et. al. [1] involves path planning for a mobile ground robot to navigate within a natural terrain. Their approach is based on learning a color-to-cost mapping that transforms a raw image captured using a front-mounted monocular camera to a cost map, that relates to the traversability of a path. A similar approach is used by Wei et. al. [2] to learn a color-to-cost map with a focus on energy-efficient path-planning.

Two related lines of work include view planning [3], [4], [5] and informative path planning [6], [7], [8]. View planning involves path planning for capturing views of a scene from different poses, commonly using an optical sensor, so as to optimize over a task such as scene classification [3], coverage [5] and/or 3-D reconstruction [4]. Informative path planning concerns with more generic sensing applications, where the objective is to plan paths that maximize information gain for a specific task. It is frequently used in applications like inspection [6] and monitoring [7], [8]. In this work, we look at the inverse problem that studies the effect of classification accuracy on path planning and propose classification criteria that efficiently utilize planning resources. There exists literature that explores both dimensions of the problem, vis-a-vis target detection and path planning. Hayashi et. al. [9] and Silwal et. al. [10], for instance, present the design of a strawberry harvesting robot and an apple harvesting robot, respectively. Both the works, follow a similar pipeline, on fruit identification, localization and planning for the robotic manipulator to pluck the fruit. The work by Hansen et. al. [11] on mapping weeds within a farm employs a combination of aerial and ground robots to map a sugar beet farm and develop vision based methods to detect weeds (thistles). They use a realization of genetic algorithm to compute tours for the ground vehicle. None of these works study the effect of classifier performance on planning decisions and path costs. In this work we do not aim to design a new detection algorithm or a path planning algorithm, rather we study the effect of classifier thresholds on planning costs. The contributions of this work, are as follows:

- We develop planning-aware threshold computation methods for binary classification tasks.
- Our method allows the user to input planning budget as a parameter and compute thresholds to fully utilize the budget.
- We also develop a method to compute thresholds that guarantee coverage of a certain minimum percentage of targets.
- We discuss and show the results of our method in a path planning application for an autonomous weed mower.

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

There does not exist any prior literature that focuses on the computation of classification thresholds informed by planning objectives.

Paper Organisation: We discuss preliminaries necessary to develop our methods in Section III. We discuss the application scenario and formulate the problem statement in Section IV. In Section V, we develop the planning-aware thresholding methods. In Section VI, we present experimental results of using our thresholding methods in a weed-mowing application. Section VII concludes the paper and identifies future directions.

III. Preliminaries

In this section we discuss system assumptions for our methods. We also discuss existing results from the literature that are used to develop our methods, namely, bounding the length of a tour for an autonomous robot and a performance evaluation and visualisation method for binary classifiers called ROC curve.

A. Assumptions

We make the following assumptions on the targets/objects of interest that the robot needs to visit:

1) **Uniform Spatial Distribution:** In the binary classification problem at hand, the two classes of objects are distributed uniformly at random throughout the operational area.
2) **Spatially Uncorrelated Features:** Let $\mathcal{F}$ be the set of identifying features for an object of interest and X be its location vector within the environment. We assume that the value of features $\mathcal{F}$ are uncorrelated with its location, i.e.

$$P(\mathcal{F}, X) = P(\mathcal{F})P(X) \tag{1}$$

It is common in robotic applications like coverage, surveillance, sensing and/or data collection, for points of interest to be distributed randomly in an area and for the robot to visit them. Also, in cases where the spatial distribution is unknown, it is common to assume a uniform random distribution in the environment.

The assumption on spatially uncorrelated features represents a large class of robotic coverage applications that involve visits to sensing and actuating sites whose spatial location is independent of their identifying features. Consider for instance, an application that involves detecting apples on a tree and plucking them using a robotic arm. The spatial distribution of apples on a tree is random, and their identifying features (color, shape etc.) are not correlated with their location. Another related application is path planning for aerial robots to search for and take aerial images of objects of interest like human survivors or disaster prone building. The identifying features of humans or buildings, like construction material used on rooftops, are independent of their relative locations.

B. Tour Length Bounds

To compute the cost of a robot's path to visit the objects of interest, we recall bounds [12] on the length of an optimal tour for a Traveling Salesman Problem (TSP).

Theorem 1: [12] Let $p_1, p_2, \ldots, p_m$ be a set of m independent and uniformly sampled points within a bounded region of area A in $\mathbb{R}^2$, and $\mathcal{T}(m)$ be the length of an optimal TSP tour that visits m points. Then there exists a universal constant Γ such that:

$$\lim_{m\to\infty} \mathcal{T}(p_1, p_2, ..., p_m) = \Gamma\sqrt{mA}$$

with probability $\sim$1, where

$$\Gamma_- = 0.628 \leq \Gamma \leq 0.891 = \Gamma^+.$$

C. ROC Curve

The Receiver Operating Characteristic (ROC) curve is a visualization of the performance of a binary classifier. It plots *True Positive Rate (TPR)* on the y-axis against *False Positive Rate (FPR)* on the x-axis. Data points on the curve are computed by varying the classification threshold. An efficient algorithm to generate ROC curves with time complexity $\mathcal{O}(n \log n)$ is given in [13]. By definition, ROC curve is a monotonically increasing curve, since any object classified as positive at a given threshold will continue to be classified positive at all lower threshold values. Also, in standard form, positive instances are scored higher on the thresholded metric (a simple tranformation can be added otherwise) and hence the threshold value decreases with increasing value of the x-coordinate.

IV. Problem Formulation

Consider a rectangular environment $\mathbb{E}$ of size $\mathcal{A}$ that comprises of two classes of objects, a and b, spread uniformly within the environment. Let a be the set of positive instances or targets. Given, an autonomous mobile robot with a limited operational range, we need to design a tour for the robot to visit the objects of class a within $\mathbb{E}$. Without loss of generality, we express path cost and operational range for the robot in terms of path length.

We use a square grid discretisation of the environment whose resolution is an input-parameter and is affected by the footprint of the robot. A grid cell is marked as covered if the robot visits its center. Let n be the total number of objects within $\mathbb{E}$ and ρ be the fraction of objects of class a. Then, $n_a = \rho n$, and, $n_b = (1-\rho)n$, is the number of objects of class a and class b, respectively. Let $\mathcal{C}$ be a classifier that uses features in $\mathcal{F}$ to identify objects of class a and $n_a^{'}$ be the number of positively labelled objects. Let the TPR and FPR, for $\mathcal{C}$ be α and β, respectively. Then,

$$n_a^{'} = \alpha n_a + \beta n_b \tag{2}$$

It follows from (1) and the assumption of uniform spatial distribution that a classifier that uses features in $\mathcal{F}$, satisfies (3).

$$P(\mathcal{L}, X) = P(\mathcal{L})P(X) \tag{3}$$

where, $\mathcal{L}$ is the label assigned to an object by the classifier. It is then reasonable to conclude that the classifier does not introduce any spatial correlation and the set of objects detected by the classifier follow a uniform spatial distribution. Using Theorem 1, we observe that the cost (length) of the tour for a robot can then be estimated based on the number of objects it must visit without regard to their actual location within the environment. If $\mathcal{T}(n)$ is the *cost* of an optimal tour for the robot to visit n objects, then

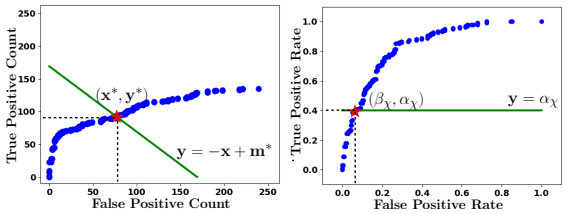
$$\mathcal{T}(n) \geq \mathcal{T}(n_a) \geq \mathcal{T}(\alpha n_a)$$

and the cost penalty, κ, on the robot's path to visit αn_a objects by virtue of misclassified instances is defined as:

$$\kappa = \frac{\mathcal{T}(n_a^{'}) - \mathcal{T}(\alpha n_a)}{\mathcal{T}(\alpha n_a)} \tag{4}$$

Then, given a robot with a maximum operational range, $\mathcal{T}^*$, that operates on a rectangular area of size A sq. meters comprising of n objects distributed uniformly in the pasture with a density ρ of instances of objects a, we address the following two problems:

- *Problem 1:* Find the maximum number of objects of class a that the robot can cover within $\mathcal{T}^*$.
- *Problem 2:* Find the minimum penalty that the robot must pay in terms of length/cost of the tour, to guarantee the coverage of at least $\chi\%$ objects of class a, where χ is an input parameter.



(a) (b)

Fig. 1: (a) Scaled ROC curve (blue circles) showing true positive count and false positive count. Green line corresponds to the available planning budget. Sum of x and y coordinates of a point on the curve corresponds to the number of positively labelled objects. The intersection point (red star) marks the required threshold value. (b) The intersection of ROC curve (blue circles) and line representing minimum coverage requirement (solid green line), marked with a red star corresponds to the required threshold value.

V. Solution Approach

The solution method comprises of two phases. First phase, is aimed at identifying and locating targets within the environment. Second phase, involves path planning for the robot to visit and service the targets. A binary classifier must label each object as a or b. The effect of a misclassified instance can result in increase in the cost of the robot's tour. Hence, the selection of an appropriate decision boundary/threshold within the binary classifier is of vital importance. In this section, we develop planning-aware threshold computation methods for the problems formulated in Section IV.

Problem 1: Let m^* be the total number of targets the robot may visit within the budget $\mathcal{T}^*$ on the length of the robot's tour. It can be claimed that:

$$\Gamma_- \sqrt{m^* A} \leq \mathcal{T}^* \leq \Gamma^+ \sqrt{m^* A} \tag{5}$$

By rearranging (5) we can bound the value of m^* as follows:

$$m_- = \left(\frac{\mathcal{T}^*}{\Gamma^+ \sqrt{A}}\right)^2 \leq m^* \leq \left(\frac{\mathcal{T}^*}{\Gamma_- \sqrt{A}}\right)^2 = m^+ \tag{6}$$

Then $m^* = m^+$ is the absolute maximum number of objects the robot can visit within the available budget.

Consider again, the binary classifier $\mathcal{C}$. To compute the maximum fraction of class a objects that the robot can cover within the given budget, we must compute a suitable threshold for the classifier. As a first step we construct the ROC curve for classifier $\mathcal{C}$. In the second step, we add a transformation to x and y axes values by multiplying with n_b and n_a, respectively. Then the x and y axes values represent false positive count and true positive count, respectively (Figure 1a). While, the transformation alters the aspect ratio of the original ROC curve, it preserves the monotonicity property. The number of instances, m_t, labeled as a at a given threshold, t, by the classifier can directly be computed as the sum of x and y coordinates of the corresponding point on the transformed ROC curve. Thus,

$$l_1 : m_t = x_t + y_t \tag{7}$$

where, x_t and y_t are the x and y coordinates, respectively, of the point on the transformed ROC curve corresponding to threshold t. We note that, equation (7) also defines a line in x-y plane that intersects with the transformed ROC curve in exactly one point (since the transformation preserves monotonicity), marked in Figure 1a with a red star. By substituting $m_t = m^*$ in (7), we find the point of intersection, (x^*, y^*) and the associated threshold value, t^*. Let α^* and β^* be the TPR and FPR values corresponding to t^* in the original ROC curve. Then α^* is the absolute maximum fraction of positive instances of class a that the robot can visit within the planning cost budget of $\mathcal{T}^*$ when using the classifier $\mathcal{C}$ to detect targets in the environment.

Problem 2: Let χ be the required percentage of targets that must be covered by the robot and α_χ be the corresponding true positive rate. To find the threshold value corresponding to minimum cost penalty, we compute the intersection of the ROC curve with the line $l_2 : y = \alpha_\chi$, as shown in Figure 1b. There may exist more than one intersection points since the ROC curve is *not strictly monotonic* and l_2 marks a horizontal line on the 2D plane. While selecting the intersection point with the biggest x-coordinate value would ensure the coverage requirement is met, it can significantly underestimate the threshold, resulting in large number of false positives (since the slope of ROC curve tends to decrease as the threshold decreases). We find the intersection point with the smallest x-coordinate value. Let β_χ be the value of the x-coordinate of the intersection point and t_χ be the associated threshold value (refer Figure 1b).

To compute the cost penalty in the robot's path, we recall relations (2) and (4). Then, the number of objects, n_a^χ, labeled as a by the classifier is:

$$n_a^\chi = \alpha_\chi n_a + \beta_\chi n_b$$

and the penalty, κ_χ, is computed as,

$$\kappa_\chi = \frac{\mathcal{T}(n_a^\chi) - \mathcal{T}(\alpha_\chi n_a)}{\mathcal{T}(\alpha_\chi n_a)} \tag{8}$$

κ_χ is the minimum cost penalty in the cost of the robot's path to visit at least χ fraction of true positive instances when using classifier $\mathcal{C}$ to detect targets in the environment. It is expressed as a fraction of the cost of an optimal tour for the robot to visit all of the true positives instances.

VI. Path planning for an autonomous mower

We implemented the approach given in Section V to compute paths for an autonomous weed mower operating on an agricultural farm. In a typical dairy pasture, weeds must be mowed after the cattle finish grazing. This is because cattle selectively eat the soft grass and leave behind patches of weeds and mature grass. The autonomous mower must visit and mow weed patches on the pasture. We present the implementation and results for planning-aware threshold computation methods on real and simulated pastures.

A. Experimental Results

We report results on the use of our methods on real agricultural data. We collected aerial images of a dairy pasture (30 m $\times$ 32 m) in the city of Morris, Minnesota using a DJI Phantom 4 drone at an altitude of 5 meters (Figure 2a). Weeds on the pasture were hand-labeled using tarpaulin strips to collect ground truth (Figure 2b). We use the model of an autonomous weed-mower with a coverage footprint of 1m $\times$ 1m and maximum operating range of 150 m in a single charge of the battery. The length of a lawn-mower path for the autonomous mower to cover the pasture is $\sim$ 500 m. We stitched the aerial images together to build the environment image, and used a pixel density of 40 pixels per meter for our work. We find that $m^* = \frac{m_- + m^+}{2}$ (see Equation 6) is a good approximation and also gives a safety margin. To detect weeds, we compute the covariance and mean of the RGB color space for each grid cell (1 sq meter), followed by principal component analysis (PCA) to reduce the feature space to 6 (computed as number of components needed to explain 99 % variance). We train a logistic regression

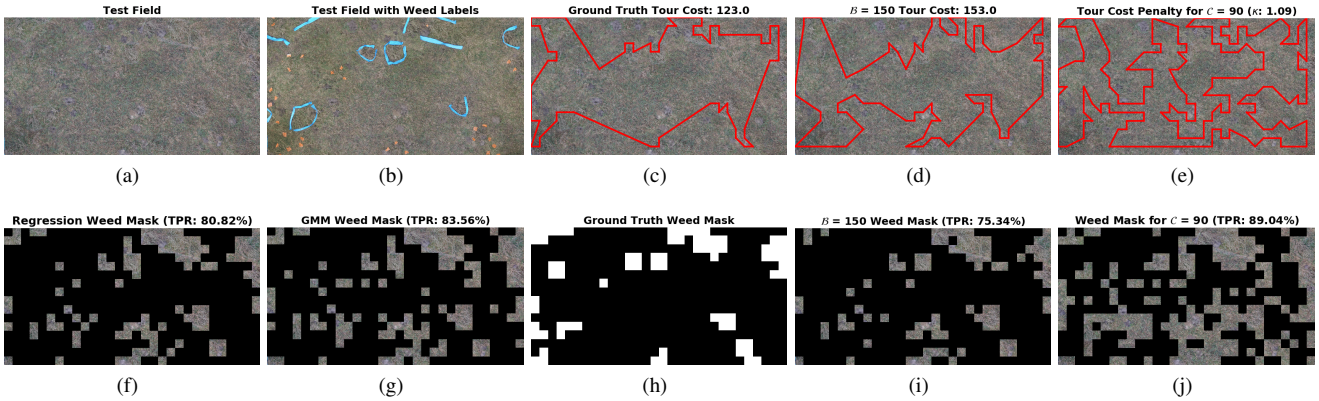


Fig. 2: (a) Stitched aerial image of the pasture. (b) Aerial image showing ground truth markers. (c) Mower tour and cost to visit actual weed patches (using ground truth). (d) Mower tour computed using the travel budget (150 meters) as a parameter. (e). Mower tour to visit weed patches for minimum coverage requirement of 90% with tour cost penalty $\kappa = 1.09$. (f). Weed mask computed using logistic regression's inherent threshold (0.5). (g). Weed mask computed using GMM based threshold to minimize misclassification error. (h) Ground truth mask for weed cover on the pasture. (i). Weeds detected on the pasture for travel budget of 150 meters for the mower. (j). Weeds detected on the pasture to match the coverage requirement of 90%.

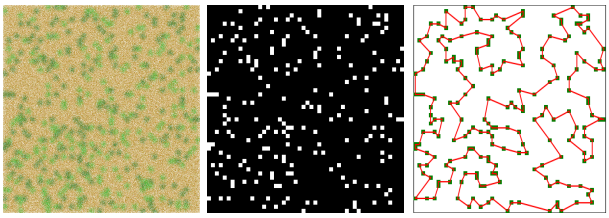


Fig. 3: A synthetic aerial image for a pasture showing weed and grass patches. (a) Pasture with $P = 16\%$ and $\rho = 0.5$. (b) Ground truth weed mask. (c) A tour for the autonomous mower to visit the weed patches.

model on the six features to compute the prediction probability with a 50:50 train-test split. The planning-aware threshold computation methods are then used to compute the decision boundary. Figures 2c and 2h show the mower tour to visit true weed patches and the corresponding ground truth weed masks, respectively.

We performed two sets of computations. The first one uses the operational range of the mower to compute the threshold. We observe that the resultant mower tour length is 153 m (Figures 2d and 2i) and allows the mower to cover 75.34% of the total weed cover. The second one computes the threshold using minimum coverage requirement as the parameter (Figures 2e and 2j). For $\mathcal{C} = 90\%$, the corresponding threshold allows the mower to cover $\sim$ 89.04% of the actual weeds. The errors are attributed to the small scale of the experiment that enhances the effect of local difference in weed density on the pastures. The tour length penalty value for the mower to cover 90% of all weeds on the pasture is $\kappa = 1.09$ and implies that the mower would need to have twice the travel budget due to the false positives. This information allows informed mission planning and resource allocation. We also show the results of using standard thresholding methods in Figures 2f and 2g. Figure 2f shows the weed mask computed using a logistic regression with its inherent threshold (0.5) and Figure 2g shows the result of computing threshold to minimize misclassification error using a Gaussian mixture model (GMM).

Parameter Name	Range of Values
Total % patches within grid cells (P)	{16%, 32%, 48%}
Weed density (ρ)	{0.4, 0.5, 0.6, 0.7, 0.8}
Planning Budget ($\mathcal{B}$)	{20%, 30%, 40%, 50%}
Min. Coverage Requirement ($\mathcal{R}$)	{40%, 50%, 60%,70%, 80%, 90%, 100%}

TABLE I: Range of simulation parameters' values.

B. Simulation Results

We also show our results in simulation while varying the value for various parameters. We generate synthetic aerial images for dairy pastures that comprise of a combination of weed and grass patches. Size of the image was fixed at 1000×1000 pixels with a resolution of 1 meter. A square grid with cells of size 20×20, was superimposed on the image resulting in 2500 grid cells. In keeping with the uniform spatial distribution assumption, weed and grass patches were distributed uniformly at random throughout the pasture. A patch (of weed or grass) corresponds to one grid cell. RGB color values for the two type of patches were sampled from two different normal distributions. The total number of patches (P) and the density of weed patches (ρ) were varied as simulation parameters. The range of values for each of the parameters is listed in Table I. We generated 20 images for each combination of P and ρ for a total of 300 instances. A sample image and mower tour to visit the weed patches on the pasture is shown in Figure 3.

We compute a score, s, for each grid cell, as the mean value of the function $256 - max(0, G - R)$ for pixels in the cell, where G and R refer to the green and red channel values, while discarding the values in the first and last quartiles. A path for the autonomous mowing robot to visit each grid cell on the pasture costs $50k$ units. We vary the planning budget, $\mathcal{B}$, for the robot to visit weed patches as percentage of the total coverage cost between 20% and 50%. The minimum coverage requirement, $\mathcal{R}$, is varied from 40% to 100%.

Results: The problem of labeling each patch on the pasture as weed or grass, affects the mission cost and objective. As a first step, we use a k-means clustering with $k = 2$ to detect all of the

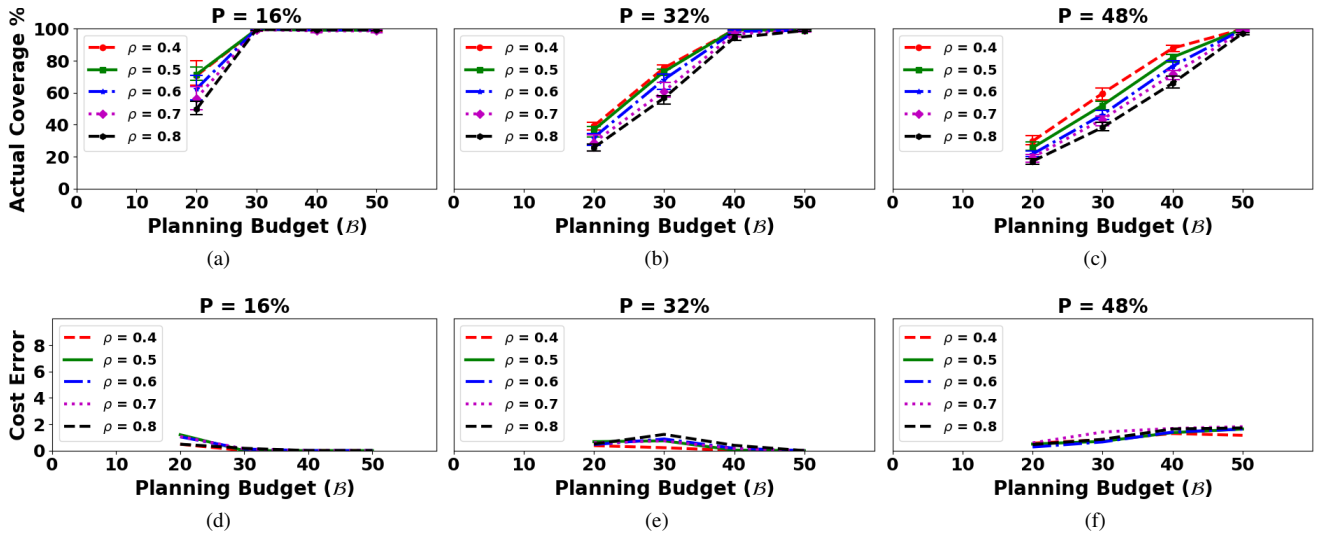


Fig. 4: Figures (a), (b) and (c) show plots for percentage of weeds covered by the autonomous mower within the available planning budget using custom computed thresholds. The budget is varied as a fraction of the cost of visiting each grid cell on the pasture. Figures (d), (e) and (f) show plots for the error in path lengths for paths that exceed the available budget.

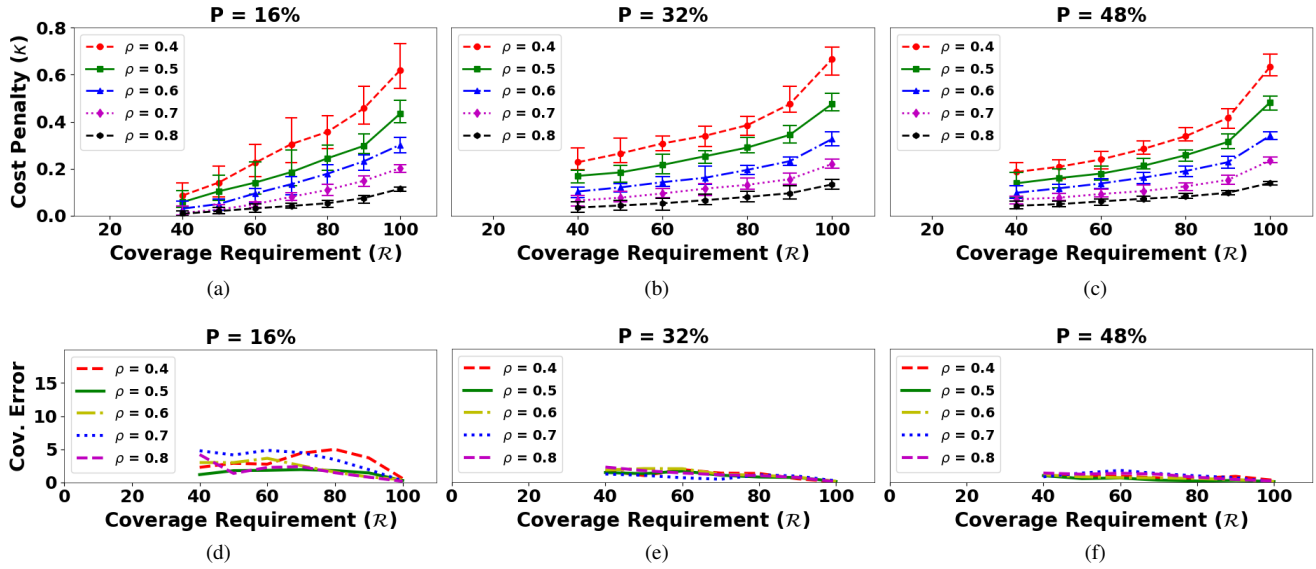


Fig. 5: Figures (a), (b) and (c) show the plots for tour cost penalty for the autonomous mower to ensure the minimum weed coverage requirement is met. Whiskers mark the first and third quartile values of the penalty, κ. Figures (d), (e) and (f) show plots for error in coverage requirement. The plots show error for instances where actual coverage is less than the coverage requirement.

patches on the pasture. We then use our planning-aware threshold computation method to compute thresholds on the score s for grid cells to identify weed patches. The thresholds for each combination of P and ρ were computed individually. Figures 4(a)-(c) show the percentage of weeds covered by the autonomous mower within the available planning budget. We compute the thresholds based on the available planning budget using our threshold computation method. As shown in the figure, our method is successfully able to utilize increase in planning budget to improve coverage. It is interesting to observe that weed coverage decreases with increases in the density (ρ) of positive instances (weeds) as well as total number (P) of instances (green patches). This is expected behavior since an increase in any of ρ or P results in an increase in the number of positive instances, thus increasing the budget requirement. We also compute and report any surplus planning costs over the planning budget, as cost error. The cost error averaged 0.57% with a standard deviation of 0.75. The cost error for different combinations of parameter values is shown in Figures 4(d)-(f).

Results for the threshold computation method using minimum coverage requirement as an input parameter are shown in Figure 5. Figures 5(a)-(c) show the penalty in the cost of the tour for the autonomous mower plotted against minimum coverage requirement. The penalty gives the fractional increase in cost to visit all of the patches labelled as weeds as compared to visiting the patches that correspond to true positives (refer equation 8). Our method allows a user to precompute the penalty and thus make decisions

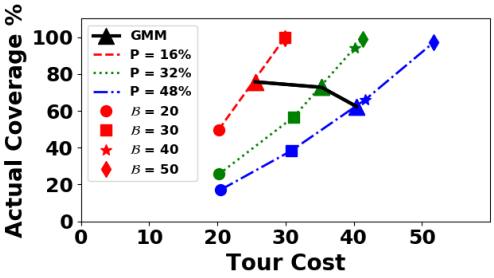


Fig. 6: Comparison for GMM-based thresholds with the proposed planning-aware threshold computation. Black solid line and triangles represent GMM data points. Other markers correspond to $\mathcal{B}$ values and colors & line-style mark the number of patches on the pasture. Planning-aware threshold computation method is able to improve coverage with the availability of additional resources.

on fuel requirement and/or other equipment. It is observed from the figure that the rate of increase in cost with the increase in coverage requirement, decreases for higher values of ρ. This result is consistent across different values of P. In the case of a higher density of positive instances (large ρ) in the operational region, the tour for the autonomous robot is more spread-out through the environment and needs only small deviations from the original tour to cover any new instances. Figures 5(d)-(f) quantify the coverage shortfall in the path of the autonomous mower. We measure the fraction of weeds covered and report the shortfall from the minimum coverage requirement, as coverage error. The mean coverage error over all simulation instances was found to be 1.38% with a standard deviation of 2.17. This work also holds importance in the context of traditional thresholding methods. We show the results of using GMM based thresholds that minimize misclassification error in Figure 6. Thresholds and the resultant value of planning objectives (coverage, cost) are fixed and cannot be customized to specific mission requirements. There isn't a way to utilize any extra planning budget for the mower, over the tour length computed using the fixed thresholds. Planning-aware threshold computations methods on the other hand, are able provide additional coverage from 25% to 37% as P goes from 16% to 48%.

C. The effect of scale

An important aspect of this work is the scale of sensing. Spatially distributed objects that follow a uniform distribution at a given scale, may follow a different distribution when the scale of sensing changes. In the weed mowing application, if the resolution was changed to $1cm$ per pixel, a 1000×1000 image would cover an area of $10m \times 10m$. Instead of observing the distribution of weeds and grass patches on the pasture we would possibly observe the distribution within patches, which need not be uniform. In a similar application that involves aerial monitoring of buildings in a city to identify disaster prone structures, spatial distribution of buildings at the scale of a city as compared to a single block could differ significantly. The distribution of fruits as observed in a path planning application for a robotic gripper picking fruits would also differ if scale of sensing were to change from observing fruits on a tree to observing fruits within a fruit cluster on the tree.

VII. Conclusion and Future Work

In this work we develop methods to compute decision boundaries for binary classification tasks in planning applications. Our planning-aware threshold computation methods are able to fully utilize planning budget and to ensure that certain minimum fraction of targets are visited. We show our results on a weed mowing application to plan paths for an autonomous mower. The thresholds computed by our algorithm are able to achieve the desired planning objectives and give additional information that aids in mission planning. To make use of the results developed in this paper, it is important to understand their limitations. All thresholding methods based on the ROC curve assume that the distribution of training and testing data are similar. In this context, it is important to carefully curate training data to represent test data distribution as accurately as possible. Extensions to this work involve relaxing the uniform spatial distribution assumption wherein a classification task must incorporate knowledge of the spatial distribution to optimize planning costs. Online version of the problem to trade-off classification accuracy with planning objectives is also interesting.

Acknowledgement

This project is supported in part by MN LCCMR program and the SmartFarm project supported by Tiné.

References

[1] W. H. Huang, M. Ollis, M. Happold, and B. A. Stancil, "Image-based path planning for outdoor mobile robots," *Journal of Field Robotics*, vol. 26, no. 2, pp. 196–211, 2009.

[2] M. Wei and V. Isler, "Energy-efficient path planning for ground robots by combining air and ground measurements," *arXiv preprint arXiv:1907.06337*, 2019.

[3] P. Roy and V. Isler, "Active view planning for counting apples in orchards," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 6027–6032.

[4] C. Peng and V. Isler, "Adaptive view planning for aerial 3d reconstruction," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 2981–2987.

[5] N. Stefas, H. Bayram, and V. Isler, "Vision-based monitoring of orchards with uavs," *Computers and Electronics in Agriculture*, vol. 163, p. 104814, 2019.

[6] G. A. Hollinger, U. Mitra, and G. S. Sukhatme, "Active classification: Theory and application to underwater inspection," in *Robotics Research*. Springer, 2017, pp. 95–110.

[7] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, Dec 2016.

[8] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, "Online informative path planning for active classification using uavs," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5753–5758.

[9] S. Hayashi, K. Shigematsu, S. Yamamoto, K. Kobayashi, Y. Kohno, J. Kamata, and M. Kurita, "Evaluation of a strawberry-harvesting robot in a field test," *Biosys. Eng.*, vol. 105, no. 2, pp. 160 – 171, 2010.

[10] A. Silwal, J. R. Davidson, M. Karkee, C. Mo, Q. Zhang, and K. Lewis, "Design, integration, and field evaluation of a robotic apple harvester," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1140–1159, 2017.

[11] K. D. Hansen, F. Garcia-Ruiz, W. Kazmi, M. Bisgaard, A. la Cour-Harbo, J. Rasmussen, and H. J. Andersen, "An autonomous robotic system for mapping weeds in fields," *IFAC Proceedings Volumes*, vol. 46, no. 10, pp. 217 – 224, 2013.

[12] S. Steinerberger, "New bounds for the traveling salesman constant," *Advances in Applied Probability*, vol. 47, no. 1, pp. 27–36, 2015.

[13] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.