Active Learning with Spatially Sensitive Labeling Costs

Alexander Liu
University of Texas at Austin
aliu@ece.utexas.edu

Goo Jun
University of Texas at Austin
gjun@ece.utexas.edu

Joydeep Ghosh
University of Texas at Austin
ghosh@ece.utexas.edu

Abstract

In active learning, it is typically assumed that all instances require the same amount of effort to label and that the cost of labeling an instance is independent of other selected instances. In spatially distributed data such as hyperspectral imagery for land-cover classification, the act of labeling a point (i.e., determining the land-type) may involve physically traveling to a location and determining ground truth. In this case, both assumptions about label acquisition costs made by traditional active learning are broken, since costs will depend on physical locations and accessibility of all the visited points. This paper formulates and analyzes the novel problem of performing active learning on spatial data where label acquisition costs are proportional to distance traveled.

1 Introduction

In active learning, one attempts to maximize classifier performance for a given number of labeled training points by allowing the active learning algorithm to choose which points should be labeled. It is often assumed that: 1) the cost of acquiring the label for a particular point is independent of the label acquisition cost for all other points and that 2) label acquisition costs for all points are equal.

Costs that are non-uniform and are dependent on other points chosen for labeling may arise when applying active learning to spatially distributed data. For example, for classification of land-cover using hyperspectral data [6], acquiring labels may involve traveling to a particular location and performing some sort of test such as determining the type of land at that point or collecting various samples, such as soil, water, or foliage samples, that require physical access. Traveling to this point incurs some type of cost (e.g., gas or time) proportional to the distance traveled. The distance traveled also depends on the order in which one visits the points that need to be labeled, meaning that the label acquisition cost for a particular point is dependent on other, previously visited points.

In this paper, we examine a method for incorporating non-uniform, dependent label acquisition costs into an uncertainty sampling [8] approach to active learning. We show that active learning with spatially dependent costs can be addressed by combining uncertainty sampling with techniques used to solve the traveling salesman problem (and its variant, the traveling salesman problem with profits).

2 Active learning for spatial data

As in "standard" active learning, active learning on spatial data occurs in an iterative fashion where, on each iteration i, points from some unlabeled set $\mathcal{U}$ are selected by the active learner based on some criteria, labeled by some oracle, and then placed in the labeled set $\mathcal{L}$. In this paper, we study two different but related scenarios for active learning on spatial data.

In the first scenario, on iteration i, the algorithm chooses a fixed number of points to label and provides the labeler a path to travel between the points, beginning and ending at the same "home location" on each iteration[1]. Once the labels have been obtained, the algorithm retrains on the set of labeled points $\mathcal{L}$, and the process repeats. The goal is to produce a classifier that reduces empirical errors while minimizing data acquisition costs.

The second scenario also involves a "home location" where the labeler begins and ends on each iteration. However, instead of labeling a fixed number of points on each iteration, the labeler continues to label points until some traveling budget is expended. In this paper, the traveling budget is the amount of time available for traveling, sample gathering and labeling each day.

For the sake of discussion, we will refer to the first scenario as spatial active learning with a fixed batch size and refer to the second scenario as spatial active learning with a fixed traveling budget. The two problems are essentially the same except for different constraints; the solutions to these two problems are therefore quite similar.

We will use the following notation. On the ith iteration, the algorithm selects n_i points for labeling. In the first scenario, n_i is constrained to be equal to a fixed batch size n_{fix} on each iteration. In the second scenario, n_i depends on some traveling budget which we will denote as t_{max}. The actual cost of traveling and labeling points for the ith iteration will be denoted as t_i. t_i depends on the total distance d_i traveled on the ith iteration, the speed s of the labeler's vehicle, the cost of labeling a single point c_l, and the number of points labeled n_i. In particular,

$$t_i = (d_i/s) + (c_l * n_i) \tag{1}$$

and the constraint is that $t_i < t_{max}$. We will measure t_i, t_{max}, and c_l in units of time, d_i in units of length, and s in units of length/time.

Finally, we will use uncertainty scores for each point in $\mathcal{U}$ as is done in uncertainty sampling. We will denote the uncertainty score for the jth point in $\mathcal{U}$ as $u(j)$.

2.1 Combining active learning with the traveling salesman problem.

A simple but somewhat naive approach for incorporating spatially related label acquisition costs with active learning to solve the scenario with fixed batch size n_{fix} is to do the following:

1. On each iteration, based on active learning criteria (e.g., uncertainty sampling or random point selection), select n_{fix} points to label
2. Use a solution to the traveling salesman problem (TSP) to find the shortest path from home through all of the n_{fix} selected points

We consider this method naive because all spatial information is ignored when choosing the initial n_{fix} points. While these n_{fix} points may be highly informative (e.g., as measured by their uncertainty scores), it may be less costly to select n_{fix} points which are closer to each other. We use this approach as a baseline for comparisons.

This baseline algorithm is somewhat clumsy to extend for the scenario where there is a fixed traveling budget since it is unknown how many points n_i to label on a particular iteration. In order to keep a similar baseline for both scenarios, we use this somewhat inefficient approach:

1. Based on the labeling budget t_{max}, estimate the maximum number of points n_{max} that could be labeled per iteration if one ignores traveling costs; that is, let $n_{max} = t_{max}/c_l$
2. Set $n_i = n_{max}$
3. Find the total cost t_i for labeling all n_i points while traveling along an optimal path (as found using a solution to TSP) through all n_i points, starting and ending at home
4. If $t_i > t_{max}$, set $n_i = n_i - 1$ and repeat previous step; otherwise, ask oracle for labels of current n_i points

We will refer to this baseline algorithm as "random/TSP" if random sampling is used to select points or "US/TSP" if uncertainty sampling is used to select points.

[1]This home location may correspond to where the labeler's vehicle is stored/refueled or the labeler's base of operations.

2.2 Combining active learning with traveling salesman with profits.

The traveling salesman problem with profits (TSPP) is a variant of TSP. In TSPP, each city that the salesman can travel to has a profit associated with visiting that city, and the salesman is not required to visit every city. The goal is to find an optimal path through some subset of the cities that maximizes total profit under some constraint.

Our approach for performing active learning on spatial data is to transform the problem into a traveling salesman problem with profits. On each iteration of active learning, the cities that can potentially be visited by the salesman (or labeler) are the points in the unlabeled set $\mathcal{U}$. The profit associated with visiting an unlabeled point is set equal to the uncertainty score of that particular unlabeled point. That is, $p(j) = u(j)$, where $p(j)$ is the profit for visiting the jth unlabeled point and $u(j)$ is the uncertainty score of the jth unlabeled point.

The constraint under which the salesman must travel differs for our two scenarios of fixed batch size and fixed traveling budget per iteration. In the fixed batch size scenario, the constraint is that the salesman can only visit n_{fix} points per iteration. In the fixed traveling budget scenario, the salesman can visit a variable number of cities per iteration as long as the total time required to travel along all cities and reach home is less than the traveling budget t_{max} for that iteration.

The traveling salesman with profits has been studied extensively [2], and fits quite well into the problem of active learning with spatial costs. We refer to this approach as “US/TSPP”.

2.3 A generalization of active learning with traveling salesman with profits.

Empirically, we found a variant of US/TSPP to be useful: instead of supplying all possible unlabeled points to the traveling salesman with profits algorithm, only the top m points with the highest uncertainty scores (where $m \geq n_i$) are used. The advantages of this variation on our framework is that it is computationally more efficient and does well at trading off between maximizing classification performance and minimizing distance traveled. We refer to this approach as “US/TSPP (filtered)”.

More formally, our approach is as follows:

1. Input: number of points to consider m, labeled set $\mathcal{L}$, unlabeled set $\mathcal{U}$, constraint n_{fix} (for scenario with fixed batch size) or t_{max} (for scenario with fixed traveling budget)
2. Iterate
   (a) Assign uncertainty scores $\mathbf{u}$ to all points in $\mathcal{U}$
   (b) Select the min(m, $|\mathcal{U}|$) points with the highest uncertainty scores where $|\mathcal{U}|$ is the number of points in $\mathcal{U}$; let this set of unlabeled candidate points be denoted as $\mathcal{U}_C$
   (c) Set the profit for visiting the jth point in $\mathcal{U}_C$ to the uncertainty score of that point
   (d) Select points from $\mathcal{U}_C$ and create a path through these points using a solution to TSPP that satisfies traveling constraints (either n_{fix} or t_{max} depending on the scenario)

For the scenario with fixed batch size, this method generalizes both US/TSP and US/TSPP. Note that if $m = n_{fix}$, then this approach reduces to US/TSP, while if m is equal to the number of points in $\mathcal{U}$, then this approach reduces to US/TSPP.

3 Experiments

3.1 Solving TSP and TSPP

In this section, we discuss heuristics for solving the traveling salesman problem and traveling salesman problem with profits in more detail. These specific algorithms are necessary for understanding our experiments. However, our solutions do not require a specific solution to TSP or TSPP, so other approaches can be readily substituted within the framework proposed in this paper.

The traveling salesman problem has been widely studied[7], and many solutions have been proposed. We will use the following terminology and notation. Each city that the salesman travels to is denoted as a node $v(i)$, where $v(i)$ is the ith city. In addition, the salesman must start and end at some home location $v(0)$. Finally, the distance between two nodes i and j is denoted by $d(i,j)$, where $d(0,j)$ would represent the distance from the "home" location to the jth point.

We use the following heuristic, chosen because the solution is quite fast and also because of readily available code[2]. Our heuristic for solving the traveling salesman problem works as follows. The algorithm begins by randomly initializing a path through all points. The algorithm then proceeds to try to improve the path by repeatedly using 2-opt and 2.5-opt. One problem with this heuristic is determining when to terminate the algorithm. We found through preliminary experiments that setting the max number of iterations to 10,000 was quite fast for a moderate number of nodes (i.e., less than 100) and resulted in convergence to a good solution.

We use a variant of the H2 heuristic originally presented in [3] to solve the traveling salesman with profits problem. Our modified algorithm is as follows:

1. Initialization: initialize path to consist of $v(0)$ (i.e., the home node) and the point j that minimizes $\frac{d(0,j)+d(j,0)}{p(j)}$ where $p(j)$ is the profit obtained for visiting node j
2. Iterate until the number of nodes (not counting $v(0)$) in the path is equal to n_{fix} (fixed batch size constraint) or until the total traveling budget exceeds t_{max} (fixed budget constraint): Add point that minimizes $\frac{d(i,j)+d(j,k)-d(i,k)}{p(j)}$ for some point $v(j)$ not in current path and consecutive points $v(i)$ and $v(k)$ in path
3. Optimize path using solution to TSP described above

As with our method for solving the traveling salesman problem, this algorithm was chosen since it was simple to implement and because it was fast enough for experimentation.

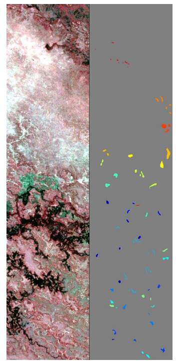
3.2 Datasets.

Land cover classification by hyperspectral image (HSI) data analysis has become an important part of remote sensing research[6]. The proposed method was evaluated on hyperspectral images taken from two geographically different locations: NASA's John F. Kennedy Space Center(KSC) and the Okavango Delta in Botswana. We will call the two datasets the KSC and Botswana datasets, respectively. The goal of these datasets is to correctly classify types of land. The KSC dataset, containing 13 classes, consists of 176 bands (i.e., features) in a hyperspectral image of size 512×614 pixels with 18m spatial resolution. The Botswana dataset, which contains 14 classes, consists of 145 bands in an image of size 1476×256 pixels with 30m spatial resolution. Figure 1 shows images of Botswana and KSC with their corresponding class maps. The grey areas in the figure denote areas which have "no label" while remaining colors correspond to particular class labels. As can be seen in this figure, only a small fraction of the entire region actually has class labels due to the expense of creating a fully labeled dataset. KSC map has only 5211 labeled points out of 314368 pixels, and Botswana map has 3248 labeled out of 377856 pixels. Aside from their use in preprocessing (see discussion of max-cut below), areas without labels are ignored during experimentation.

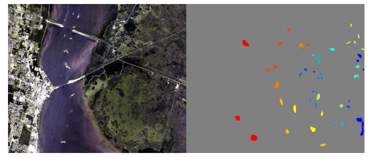
We preprocess the data in two ways that are known to be effective for classifying hyperspectral data. First we utilized spatial information in addition to the spectral information, via the max-cut algorithm described in [1], where a pixel's feature vector is augmented with features from neighboring pixels whose spectral features are similar to the pixel of interest. The max-cut algorithm takes advantage of unsupervised information and provides a way to identify pixels that are close both in physical and spectral spaces, and produces more accurate and stable classification results for spatial data. Second, we applied best-basis feature reduction which exploits the high correlation between certain adjacent spectral bands, and is tailored for hyperspectral data analysis [5].

In each dataset, each data point has a location as determined by two coordinates. We use the Euclidean distance between coordinates as the distance required to travel between two points.

[2]Our experiments were run using a modified version of the code used in the Matlab demonstration of the traveling salesman problem (travel.m)



(a) Botswana



(b) KSC

Figure 1: Botswana and KSC images with class maps

3.3 Experimental setup.

We use LDA as the classifier in our experiments and use uncertainty scores $\mathbf{u}$ inversely proportional to the margin between the largest posterior probability and second largest posterior probability as predicted by LDA. We have also run several experiments on two-class problems using both LDA and SVMs, where uncertainty scores for SVMs are inversely proportional to the distance of a point from the separating hyperplane. Trends for two-class problems for both LDA and SVMs are similar to trends for the full, multiclass problem using LDA, so we omit these results for sake of brevity.

We partition the data into training and test sets using five runs of ten-fold cross-validation and average the results. In particular, we use stratified sampling such that each fold has the same proportion of points from each class. Once the training and test sets have been created, 5% of the training data is randomly selected and labeled to form the initial labeled set $\mathcal{L}$, the remaining unlabeled training data is placed in $\mathcal{U}$, and a "home" location is randomly selected from $\mathcal{L}$. We use the best-basis feature reduction method to create a set of 120 features.

We run two sets of experiments. In the first set of experiments, we address the scenario where a fixed number of points are labeled on each iteration. In these experiments, the number of points n_{fix} labeled on each iteration is set to 10 points, and labeling costs are measured in terms of distance. In the second set of experiments, we address the scenario with fixed traveling budget. In this scenario, labeling costs are measured in units of time, where the constraint on each iteration is that the time t_i spent traveling and labeling points on the ith iteration is less than t_{max}. As mentioned, t_i depends on the vehicles average speed s and the cost for labeling a single point c_l. Experimentally, we tried a variety of values for s and c_l, but found that specific values do not seem to affect general trends much. Because of space constraints, we present results only for the case where $s = 80.5$ kilometers per hour[3], $c_l = 10$ minutes, and $t_{max} = 8$ hours. Finally, US/TSPP(filtered) requires a parameter m to determine the number of points in $\mathcal{U}_C$, which we simply set to 100 in all experiments. Although we did not investigate this, additional tuning of m may be useful.

In general, the execution time of US/TSPP is much slower than US/TSP or US/TSPP(filtered). For much larger datasets, it is possible that US/TSPP may become too slow. However, US/TSPP(filtered) will not have scaling issues if one keeps the number of filtered points low. Regardless, in the problem setting being considered, the time required to label points is much larger than the time required to execute the active learning algorithm, so a detailed study on execution times is outside the scope of this paper.

4 Results

Results are presented in figure 2 and table 1. For both scenarios (fixed batch size and fixed traveling budget), the general trends and shapes of the curves in the results are quite similar.

[3] i.e., roughly 50 miles per hour.

Table 1: Averages and standard deviations for specific points for scenario with fixed batch size (top two tables) and scenario with fixed traveling budget (bottom two tables)

dataset	activeLearner	errorRateAt50KM	errorRateAt100KM	errorRateAt250KM
ksc	random/TSP	0.1231 (0.0160)	0.1143 (0.0150)	0.1000 (0.0154)
ksc	US/TSP	0.1114 (0.0130)	0.0954 (0.0954)	0.0688 (0.0126)
ksc	US/TSPP(filt)	0.1032 (0.0157)	0.0870 (0.0124)	0.0606 (0.0118)
ksc	US/TSPP	0.0974 (0.0138)	0.0868 (0.0135)	0.0701 (0.0141)
bots	random/TSP	0.2555 (0.0376)	0.2022 (0.0333)	0.1215 (0.0225)
bots	US/TSP	0.2447 (0.0391)	0.1852 (0.0311)	0.0970 (0.0176)
bots	US/TSPP(filt)	0.1471 (0.0373)	0.0990 (0.0246)	0.0513 (0.0182)
bots	US/TSPP	0.0947 (0.0247)	0.0733 (0.0215)	0.0521 (0.0189)

dataset	activeLearner	errorRateAt25 hours	errorRateAt50 hours	errorRateAt100 hours
ksc	random/TSP	0.1137 (0.0150)	0.1015 (0.0127)	0.0911 (0.0119)
ksc	US/TSP	0.0956 (0.0114)	0.0734 (0.0079)	0.0515 (0.0094)
ksc	US/TSPP(filt)	0.0886 (0.0086)	0.0655 (0.0082)	0.0465 (0.0087)
ksc	US/TSPP	0.0912 (0.0142)	0.0752 (0.0154)	0.0592 (0.0153)
bots	random/TSP	0.1353 (0.0270)	0.1268 (0.0235)	0.0783 (0.0112)
bots	US/TSP	0.1292 (0.0187)	0.1109 (0.0155)	0.0557 (0.0127)
bots	US/TSPP(filt)	0.1150 (0.0257)	0.0689 (0.0150)	0.0363 (0.0130)
bots	US/TSPP	0.0760 (0.0250)	0.0590 (0.0169)	0.0450 (0.0147)

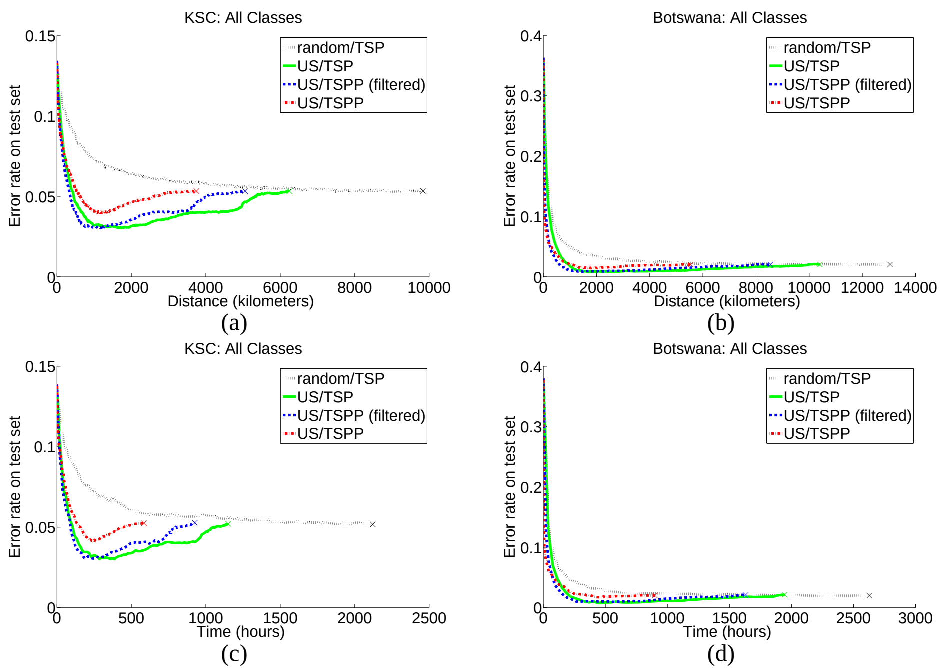


Figure 2: Average results; top row contains results for scenario I (fixed number of points per iteration) while bottom row contains results for scenario II (fixed budget per iteration)

In the graphs in figure 2, we have modified the traditional "banana curve" format of active learning in order to take into account different label acquisition costs. In this case, the horizontal axis is proportional to the total cost (e.g., distance traveled or time expended) needed to label $\mathcal{L}$. Thus, a point on the curve represents the cost expended by a labeler in order for the classifier to achieve a certain level of performance as measured by error rate on the test set. A number of interesting observations can be drawn from this type of figure. Unlike a traditional curve in active learning, the right-most point of each curve on each graph will be at a different place. This point corresponds to where all the points originally in $\mathcal{U}$ have been labeled (we have denoted this point with an "X" in the graphs for ease of visibility). The fact that different curves end at different places indicates that each method requires different amounts of effort to label all of $\mathcal{U}$.

(a) random/TSP (b) US/TSP (c) US/TSPP (filtered) (d) US/TSPP

Figure 3: Example paths for each of the methods in our experiments

Based on this right-most point, in terms of minimizing the total cost to label all of $\mathcal{U}$, US/TSPP outperforms US/TSPP (filtered) which outperforms US/TSP which outperforms random sampling on both datasets for both scenarios. This shows that our proposed approach, US/TSPP always reduces total labeling effort compared to US/TSP when labeling all of $\mathcal{U}$, and that US/TSPP (filtered) lies somewhere in-between. In many cases, this reduction is quite large. For example, the cost for labeling all of $\mathcal{U}$ using US/TSPP is, on average, less than half the cost if using US/TSP. For the sake of illustration, we have plotted four example paths on the KSC dataset[4], with one path for each tested method, in figure 3. Here, the distance saved by US/TSPP and US/TSPP(filtered) over random/TSP and US/TSP is also apparent.

Another question in active learning is how quickly a method reduces classification error. Looking at the graphs, one can qualitatively make the judgement that US/TSPP tends to reduce the error rate fastest. Quantitatively, one can compare the average error rate when the labeler has not yet traveled very far. In table 1, we have listed some error rates when the traveler has not expended much cost. If looking at the lowest obtainable error rate, US/TSPP(filtered) is able to reduce error rate further than US/TSPP. In general, the lowest error rate (regardless of how much effort required to reach that point) is comparable for US/TSPP (filtered) and US/TSP. Thus, US/TSPP(filtered) empirically achieves an error rate as low as US/TSP and reduces error rate more quickly than US/TSP.

Finally, in the results (particularly on the KSC dataset), we see that the error rate can be lower when only a subset of $\mathcal{U}$ has been labeled as opposed to when all of $\mathcal{U}$ has been labeled. This phenomenon has been observed in other works on active learning as well [12], where the active learner stops learning when there is no more unlabeled samples in the margin of the current SVM model. It is also possible to perform cross-validation(CV) on the current labeled set and stops the process before the CV error starts to increase. Stopping the labeling process early may be useful in reducing overall classification error, but we do not study this problem further in this paper.

5 Related work

In this paper, we take an uncertainty sampling approach to active learning. One could also take a loss reduction approach [11]. That is, instead of setting the profit of visiting a point in $\mathcal{U}$ equal to the uncertainty score for that point, one could set the profit for visiting a point equal to the expected reduction in loss for adding the point to $\mathcal{L}$. There are several benefits to using a loss reduction approach. For example, [10] uses a loss reduction approach on the KSC and Botswana datasets, so it is known that loss reduction can work well for hyperspectral data. In [9], it was shown how one can incorporate non-uniform label acquisition costs into a loss reduction type setting. However, the method for incorporating label acquisition costs in [9] is quite different than our current work, and we are unsure how one would extend the technique in [9] to incorporate dependent label acquisition costs.

One major drawback of loss reduction type approaches is computational efficiency. To obtain the expected reduction in loss for all points in $\mathcal{U}$, one needs to retrain the classifier $|\mathcal{U}| * n_c$ times, where n_c is the number of classes. This is particularly problematic for our hyperspectral datasets, where KSC contains 13 classes and Botswana contains 14 classes. While there are methods of performing loss reducing active learning more quickly [11], computational efficiency was the main reason for favoring uncertainty sampling in our current set of experiments.

[4] we arbitrarily chose to plot the 25th iteration of active learning for each method for the fixed batch size scenario.

Finally, although many active learning strategies have been proposed during the last 15 years, there exist few algorithms that consider spatial characteristics of unlabeled samples. [10] proposed an active learning algorithm for hyperspectral data that adapts a classifier for spatial variation of spectral signatures. However, it does not take into account any form of varying label acquisition costs based on spatial data. An active learning algorithm to efficiently model spatial phenomena with Gaussian process has been proposed [4], but the algorithm is used to model spatially varying quantities and is not applicable to classification problems. We are unaware of any active learning studies which take spatially dependent label acquisition costs into account.

6 Conclusion

When performing active learning on spatially distributed data such as hyperspectral data or other GIS applications, there are variable label acquisition costs involved in labeling each point. These label acquisition costs may depend on the distance the labeler needs to travel in order to label each point selected by active learning. Standard active learning techniques are unable to handle such variable costs, and assume that the cost of labeling each point is uniform and that the cost of labeling each point is independent of all other points being labeled on a particular iteration of active learning. If the cost of labeling a point is proportional to the distance needed to travel to that point, and if the labeler travels to several points on each iteration of active learning, then both of these assumptions made by traditional active learning are broken.

In this paper, we have presented methods for solving this variant of active learning where label acquisition costs are proportional to the distance required to travel to a point. We have addressed two scenarios for spatial active learning: one where there are a fixed number of points labeled on each iteration, and one where the number of points labeled on each iteration are dependent on some fixed budget. Other scenarios involving spatially related label acquisition costs are possible. In addition, our work has implications for other areas of cost-sensitive learning such as transfer learning.

References

[1] Y. Chen, M. Crawford, and J. Ghosh. Knowledge based stacking of hyperspectral data for land cover classification. *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*, pages 316–322, 2007.

[2] D. Feillet, P. Dejax, and M. Gendreau. Traveling salesman problems with profits. *Transportation Science*, 39:188–205, 2005.

[3] M. Gendreau, G. Laporte, and F. Semet. A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks*, 32:263–273, 1998.

[4] A. Krause and C. Guestrin. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 449–456, New York, NY, USA, 2007. ACM.

[5] S. Kumar, J. Ghosh, and M. M. Crawford. Best-bases feature extraction algorithms for classification of hyperspectral data. *IEEE Trans. on Geosci. and Remote Sens.*, 39(7):1368–1379, 2001.

[6] D. Landgrebe. Hyperspectral image data analysis. *Signal Processing Magazine, IEEE*, 19(1):17–28, Jan 2002.

[7] E. L. Laweler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley, 1985.

[8] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1994.

[9] D. D. Margineantu. Active cost-sensitive learning. In *the Nineteenth International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005.

[10] S. Rajan, J. Ghosh, and M. M. Crawford. An active learning approach to hyperspectral data classification. *IEEE Trans. on Geosci. and Remote Sens.*, 46(4):1231–1242, 2008.

[11] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, pages 441–448. Morgan Kaufmann Publishers Inc., 2001.

[12] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proc. 17th International Conf. on Machine Learning*, pages 839–846. Morgan Kaufmann, San Francisco, CA, 2000.