Efficient Learning in Computer-Aided Diagnosis through Label Propagation

Samuel Berglin[a], Eura Shin[b], Jacob Furst[c], and Daniela Raicu[c]

[a]University of Wisconsin, Dept. of Statistics, Madison, Wisconsin, USA;
[b]University of Kentucky, Dept. of Computer Science, Lexington, Kentucky, USA;
[c]DePaul University, School of Computing and Digital Media, Chicago, Illinois, USA

ABSTRACT

Computer-Aided Diagnosis (CADx) systems can be used to provide second opinions in the medical diagnostic process. These CADx systems are expensive to build as they require a large amount of correctly labeled example data. In order to ensure the accuracy of a training label, a radiograph may be assessed by multiple radiologists, increasing the time and money necessary to build these diagnostic systems. In this paper, we minimize the cost necessary to train CADx systems while accounting for unreliable labels by reducing label uncertainty. We introduce a method which reduces the cost required to build a CADx system while improving the overall accuracy and demonstrate it on the Lung Image Database Consortium (LIDC) database. We exploit similarities between images by clustering image features of lung nodule CT scans and propagating a single label throughout the cluster. By informatively choosing better labels through clustering, this method achieves a stronger accuracy (5.2% increase) while using fewer labels (29% less) compared to a state of the art label saving technique designed for this medical dataset.

Keywords: computer-aided diagnosis, uncertain labels, LIDC, iterative classification, selective labeling, resource allocation, cost, machine learning

1. INTRODUCTION

Early detection is crucial to the success of most lung cancer treatments. Computer-Aided Diagnosis (CADx) systems act as second readers to assist medical professionals in making quicker and more accurate diagnoses. In this study, we improve a CADx system that uses Computed Tomography (CT) scans from the Lung Image Database Consortium (LIDC) dataset.[1] In order to classify a lung nodule, supervised classifiers use features extracted from the CT scans to assign a label, or malignancy rating to a potential cancer. These classifiers require large amounts of labeled data for high classification accuracy.

There are many challenges associated with acquiring this labeled data. In reality, every label corresponds to a diagnosis from a medical expert. Acquiring enough labels to build an accurate classifier is costly in terms of time and money. Furthermore, these annotations are nontrivial and diagnoses often vary between radiologists; there is no ground truth, or golden standard, for diagnosing these nodules. As a result, for difficult cases, more than one expert may be required to assess a nodule, leading to multiple labels per case. These uncertain labels which lack ground truth are referred to noisy labels in the machine learning community.[2] In addition to the LIDC dataset, multiple labels per case also are present in idealized radiology peer review systems.[3] For example, in some peer review approaches, cases are outsourced to neutral, external third party of experts using a standardized system where multiple radiologists independently assess the same image.

The goal of this paper is to minimize the cost necessary to train CAD systems while accounting for noisy labels by reducing label uncertainty. This problem is unique in the context of active learning and uncertainty reduction because the identity of the labelers are not known, and therefore, there is no way to apply prior methods of annotator assessment to this scenario. A previous selective-iterative classification (SIC) algorithm approached

Further author information:
S.B.: E-mail: berglin@wisc.edu
E.S.: E-mail: eura.shin@uky.edu

this problem by requesting new labels only for samples that were misclassified at a previous iteration.[4] This algorithm builds an initial classification model by choosing one random radiologist label for all samples. Our paper improves on the selective-iterative method by making an informed choice at this first iteration, rather than labeling with random selection. We cluster using low-level image features of the CT scans to determine which images are most similar to one another, and therefore, most likely to obtain the same radiologist rating. We then propagate the majority label of the representative sample, or the sample at the center of each cluster. We expect that initially propagating a common label throughout groupings of similar samples will reduce unnecessary label uncertainty. By using cluster-guided label propagation, we utilize annotations as efficiently as possible to create classifiers with reduced cost and higher performance. Using less labels has the potential to increase accuracy since disregarding uninformative labels can boost classification accuracy.

2. BACKGROUND

2.1 Reduction in Label Uncertainty

There has been extensive research to reduce label uncertainty in labeled datasets where the classification task is inherently subjective, resulting in uncertain reference truths. This is the case when labels rely on human interpretation rather than on quantitative analysis, such as lab tests and pathology reports. For example, Sheng et al. acknowledge that repeated labeling does not directly improve the quality of classification models, instead placing emphasis on the importance of selective labeling.[2] Furthermore, Jin and Gharahmani have shown that a selective EM model for finding correct labels produces better classifiers than a naive, unselective approach.[5] This is especially relevant in the field of medicine, where diagnoses may vary even amongst expert opinions. Mavandadi et al. improved the diagnosis of red blood cells potentially infected with malaria by training a model that incorporates an error probability associated with each expert labeler.[6] Similarly, Raykar et al. propose an algorithm that evaluates the label performance of different experts while simultaneously learning the classifier and the true label.[7] While these contributions address the issue of building classifiers on disagreeing expert opinions, they assume access to all annotator labels and do not attempt to minimize label cost.

2.2 Cost Reduction

In the context of label cost and uncertainty reduction, Yan et al. employ a probabilistic model to automatically choose the most appropriate annotator to query in addition to choosing the most useful data points in an active learning scheme.[8] However, this method relies on the known identity of a labeler. In cases where the identity of the labelers are hidden and the quality of every label is considered equal, methods which measure annotator performance are not applicable.

More specific to the problem of efficiently reducing label uncertainty with anonymous annotators, Riely et al. use a selective-iterative approach in building a classification model to predict the malignancy of lung nodules.[4] For every image sample of a nodule, this approach chooses a random radiologist label to be used in training an initial classification model. This model is improved at each iteration; samples that are incorrectly labeled are given an additional label and the model is re-trained using this new information. This process simulates asking for additional labels when the predicted label does not agree with existing data. Overall, it leverages known labels to determine which cases labels are uncertain.

Our method builds on this selective-iterative method by improving the quality of the labels used to train the classification model at the first step. Therefore, we use the selective-iterative approach as our point of comparison, with the goal of building a classifier that will have better accuracy and use fewer labels than the original method.

2.3 Representative Samples from Clustering

Since we are interested in using fewer labels, we rely on low-level image features of a CT scan to infer representative points for feature space neighborhoods. In previous work, Nguyen and Smeulders have used pre-clustering to guide the active learning process.[9] The authors extracted representative samples from clustered data and used these samples to refine a classification decision boundary. Along the same line, Dasgupta and Hsu explored a hierarchical clustering scheme to identify a subset of points that are a good representation of the entire dataset.[10]

We use the idea of representativeness within clusters from past work to form clusters that are expected to bear similar malignancy ratings. This is fundamentally different from prior work, as we are using clusters to reveal similarities within the data rather than to pinpoint the most informative points to query. By propagating the label of the representative sample to the remaining points in each cluster, we expect to diminish label uncertainty and reduce cost in the first stage of the SIC process.

3. METHODOLOGY

3.1 Data

The LIDC dataset (publicly available at ncia.nci.nih.gov) is a collection of Computer Tomography (CT) scans with annotated lesions and accompanying labels from up to four radiologists.[1] This dataset has two notable characteristics. The first characteristic is multi-class labels. These malignancy ratings range from 1 to 5 correspond to: 1. Highly Unlikely, 2. Moderately Unlikely, 3. Indeterminate, 4. Moderately Suspicious, 5. Highly Suspicious.

The second characteristic is uncertain labels. Each nodule image has up to four labels, where each label corresponds to a different radiologist rating. There is variability in these labels because there was no forced consensus among radiologists when annotating the samples. In addition, only a fraction of these images have follow up biopsy data on the true malignancy of a nodule. Because the absolute truth for these labels are not given, no set of labels is necessarily correct; they may only be used to formulate a reference truth. A reference truth is the label currently regarded as correct for a given image. It is derived from the images currently known labels. That is, if an image has four labels it will be formed from all four. Likewise, if only two labels are known it will be formed from those two.

Because our methodology is based on an iterative approach, we illustrate it using a subset of the LIDC dataset consisting of the 810 samples for which all four radiologists provided a malignancy rating.

3.2 Image Features

We used 64 features extracted from each image to stay consistent with prior work by Riely et al.[4] Extracted features can be divided as follows: shape, size, intensity, and texture. The shape features include compactness, eccentricity, circularity, roughness, solidity, extent, elongation, and standard deviation of radial distance. The size features are area, convex area, perimeter, convex perimeter, equivalent diameter, major axis length, and minor axis length. The intensity features are the minimum, maximum, mean, and standard deviation of the gray level intensity of every pixel in both the segmented nodule and its background. We also used the absolute value of the difference between the means of the nodule and background intensities, denoted as intensity difference. Finally, the texture features consisted of eleven Haralick features, five Markov features, and twenty-four Gabor features.

3.3 Label Processing

The five rating classification problem is unbalanced. The mode label (mode of four ratings) class distribution is displayed in Figure 1. The five possible malignancy labels of each nodule were mapped to three labels to balance the dataset as in Reily et al. We mapped labels $\{1, 2\} \rightarrow \{1\}$, $\{3\} \rightarrow \{2\}$, and $\{4, 5\} \rightarrow \{3\}$. Semantically, these new labels correspond to: 1. unlikely, 2. indeterminate, and 3. suspicious. We test our approach on both the balanced and unbalanced datasets.

The LIDC dataset contains significantly more ratings of 3 for malignancy than any other value. Mapping the labels is intended to balance this dataset. SIC and the label propagation algorithm are both also tested on the original dataset with no mapping in the results section.

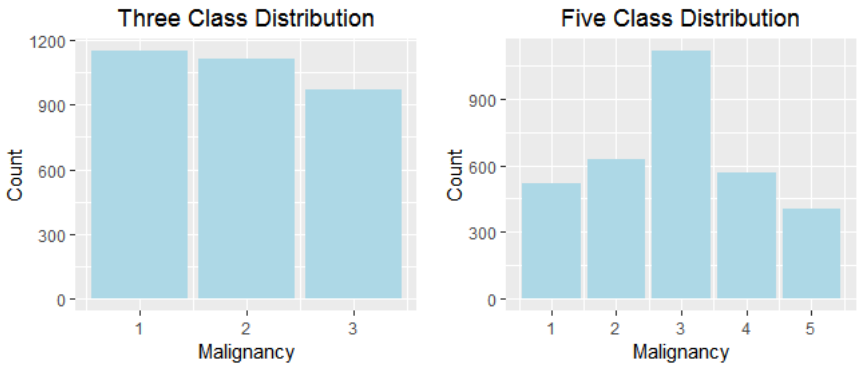

Figure 1. Transformation from unbalanced 5-class problem to balanced 3-class problem.

3.4 Selective Iterative Classification

To understand our contribution, we first discuss the selective iterative classification algorithm developed in Reily et al. It can be succinctly described through pseudo-code.

Algorithm 1: Selective Iterative Classification

```
for each n in 1:N cases do
    Shuffle the P labels
end
Pick first label as initial reference truth
Train and store CART tree on initial reference truth
for p in 1:P possible labels do
    for n in 1:N cases do
        if Case n was misclassified at previous iteration p − 1 then
            Add new label to reference truth and take mode for new reference truth
        end
    end
    Train and store CART tree based on new labels
    Evaluate accuracy on test set
end
```

The algorithm shuffles the order of the labels to remove order bias. It then uses the first label as the initial reference truth to train a CART decision tree, the chosen supervised classifier. Although a decision tree was used, this approach works with any supervised classifier. On the next iteration, the misclassified training cases labels are readjusted to be the mode of the first and second shuffled labels. This process is repeated until all labels have been used. We denote each repetition as an iteration. Since every point in this dataset has four possible labels, there are four possible iterations. Figure 2 outlines the process of a changing reference truth that trains the classifier at each iteration.

While the classification trees are built using training data exclusively, the remaining test data is also passed through the classifier and given updated labels based on the process in Figure 2 at every iteration. This is to formulate reference truths in a fashion similar as during training. Note that if a label is propagated, the propagated label becomes the initial reference truth. However, if it is misclassified, the propagated label is discarded and that observations first label then becomes the new reference truth. Then, the standard process in Figure 2 ensues.

The purpose of this method is to save labels by only requesting more labels if the current label is not supported by the rest of the data and is misclassified. This way, the algorithm can leverage patterns found by the classifier to request labels only for difficult cases. Clearly, this saves labels when compared to asking for multiple opinions

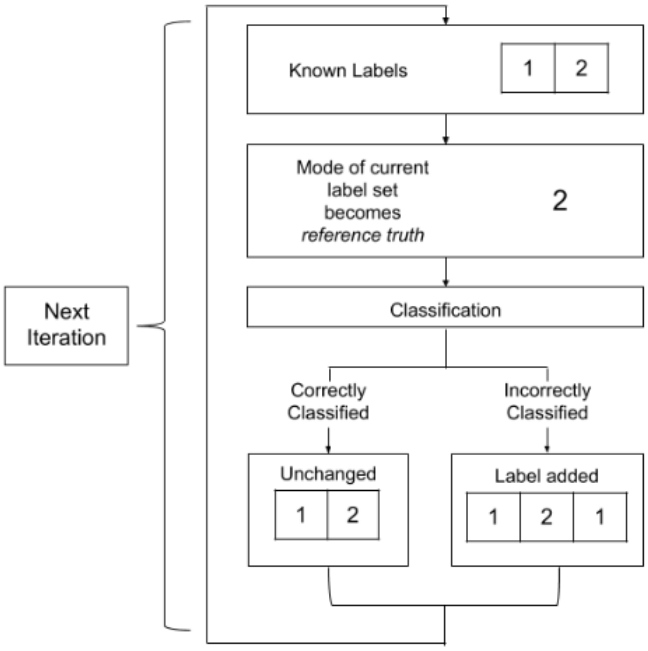

Figure 2. Diagram of progression of reference truth in SIC algorithm through an example.[4]

for every case. It also reduces label uncertainty, since asking for more annotations for cases that are already supported by prior data can introduce label uncertainty (Reily et al., 2015).

As mentioned above, any supervised classifier can be used in this process. However, for comparison with prior methods, we continue to use CART via the rpart package in R.10. We allowed trees to grow deep and then pruned according to cross validated error (within the training set). Trees were pruned to be the smallest tree within one standard error of the tree with the smallest cross validated error. In order to facilitate deep initial growth, we set the minimum number of observations in a parent node to 3 and the complexity parameter (cp) to 10^{-10}. These parameters yielded overfit trees that could be then be pruned. However, Reily et al. prevented overfitting solely by customizing stop rules during tree growth. Since the addition of pruning is an improvement but not a novel contribution, we include our results with and without it.

Furthermore, to account for bias from shuffling the labels, we repeated this process 20 times to approximate the 4! possible labels as in Reily et al. Each time, we split data into training (80%) and testing (20%), and these sets were maintained when comparing methods.

3.5 Propagated Selective Iterative Classification

Propagated SIC alters the method in which labels are assigned at the first iteration of SIC. We cluster image features to save labels at the first iteration. From there, we continue the standard SIC algorithm. Our method

is as outlined in Algorithm 2.

Algorithm 2: Propagated selective iterative classification

```
for each n in 1:N cases do
    Shuffle the P labels
end
Cluster the data
for each cluster C do
    Propagate the mode label l of the medoid of C to all other points in C and use it as initial reference truth
end
Enter SIC (Algorithm 1) line 5
```

3.6 Clustering

Our method of clustering starts with a hierarchical clustering of the data and prunes this dendrogram for the smallest clusters of at least a certain size. This is similar to the pruning method described in Dasgupta and Hsu, although size is the control parameter, rather than purity.[10] We cannot use label purity as the parameter because we wish to conserve labels. We use this size-based method to ensure all clusters are large enough to effectively save labels but small enough to ensure points within the same cluster are similar. Furthermore, plotting the data with respect to the first two principal components revealed that our data was too evenly spaced for density based clustering methods, such as DBSCAN.

In our algorithm, we begin with an existing hierarchical clustering of the data. We used average linkage to build the dendrogram bottom up. It starts at the lowest level on the dendrogram and collects clusters of adequate size as it continues making horizontal cuts up the tree. Consider Figure 3. For a size parameter of eight, the algorithm starts by considering the clusters A, D, B, C, G. The clusters B, C and A, D do not independently contain at least eight points and are replaced by the next node in the tree, resulting in a new set of clusters F, E, G. Because the algorithm does not allow a single point to belong to multiple clusters, the algorithm stops considering clusters beyond a cluster that meets the size requirement. This may result in orphan clusters such as G. Samples within these orphan clusters are considered unclustered and are given an initial random label as in the first iteration of the SIC process. This clustering method is described in Algorithm 3.

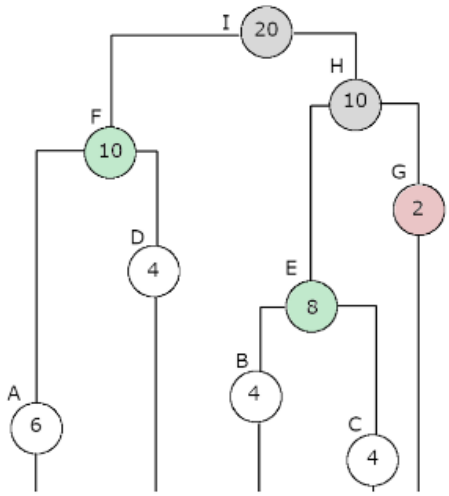

Figure 3. Example dendrogram of hierarchical clustering with algorithm results; chosen clusters in green, ignored clusters in grey, orphan clusters in red.

Algorithm 3: Size-based pruning for hierarchical clustering

input : Dataset of N points, size parameter S
output : List of clusters
initialize: $L = N$, where L is the level on the dendrogram of 1 (top) to N (bottom) levels
$U = u_1, ..., u_N$, where U is the list of points that have not been clustered

```
while |U| > 0 and L > 1 do
    cut tree at L, resulting in clusters C_1...C_L
    for C_i in C_1...C_L do
        if all points in C_i are in U and |C_i| ≥ S then
            Add C_i to the list of clusters
            U = U − C_i
        end
    end
    L − −
end
return List of clusters
```

Note that the level on the dendrogram L is only used to traverse the clusters of the dendrogram to find the smallest clusters that are larger than eight and represent unique data. Our algorithm then outputs this list of clusters. We postpone discussion of the parameter S, as it relies on our method of label propagation that is discussed next.

3.7 Label Propagation

Assume there are K clusters, $C_1, ..., C_K$. For each cluster C_i the respective medoid m_i is the closest point to the center of the cluster, which is the mean of all feature vectors of C_i. Each C_i corresponds to a subset of the data. The mode of all labels for each m_i is assigned to the remaining points in C_i. We call this process label propagation. Label propagation uses the medoid of each cluster as a representative point.

Calculation of the size parameter S for our clustering algorithm is straightforward for datasets like LIDC with a uniform number of labels for every point. We let $S = \lceil P/Q \rceil$, where P is the number of labels per point and Q is the maximum fraction of labels to be used within each cluster.

In our dataset with $P = 4$, we use $Q = 12$ to save at least half the labels for each cluster. Thus, $S = 8$. As a result, at most half the labels are necessary to label the entire cluster when compared to taking single random labels for each observation. Adjusting Q, and thus S, adjusts how assertively labels are propagated to neighbors. If Q is smaller, S increases and the final clusters are larger. Thus, labels are propagated from the cluster representative to more points. This saves labels, but decreases the representativeness of the propagated label for the points near the clusters border.

This algorithm exploits the image feature similarities between points of the same cluster to reduce the total number of labels necessary, T. For a dataset D with N observations, SIC requires N labels in the first iteration (one label for every point). For a clustering resulting in K clusters $C_1, ..., C_K$, the total number of initial labels can be described in 1 where l_x is the number of labels for a given point x and m_i is the medoid of cluster c_i. In our dataset all of our points have four labels, so $\forall x \in D : l_x = 4$.

$$T = N - |\cup_{i=1}^{K} C_i| + \sum_{i=1}^{K} l_{m_i} \tag{1}$$

The first two terms represent the number of orphan points, or points that do not belong to a cluster. For these points, one label is randomly selected to be their initial reference truth. The final term sums the number of labels used for each clusters representative.

4. RESULTS AND DISCUSSION

4.1 Cluster Analysis

The size composition of the clusters is shown in Figure 4. Note that the clustering algorithm searched for the smallest cluster of at least $s = 8$. Therefore, some clusters will have a size greater than 8.

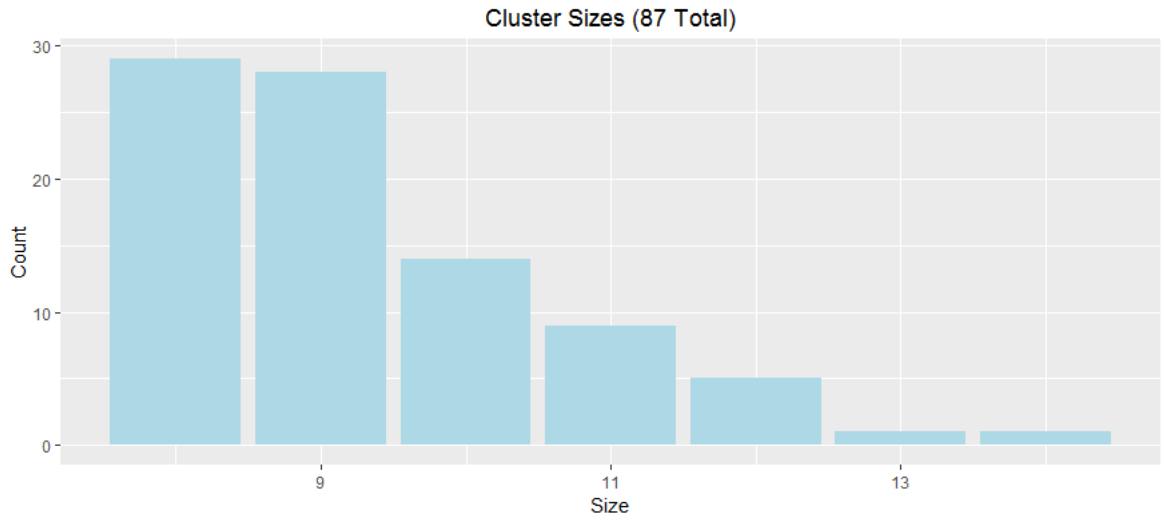

Figure 4. Size distribution of resulting 87 clusters with size parameter of 8.

Figure 5 is a comparison between the malignancy ratings assigned by mode, a randomly chosen radiologist, and the propagated labels color mapped onto the first two principal components of the feature data. This visualization represents the state of the labels at the first iteration of the SIC process compared to the goal coloration set by the mode of all radiologist ratings. Visually, the propagated labels maintain similar coloration to the mode labels while using a tenth of the cost.

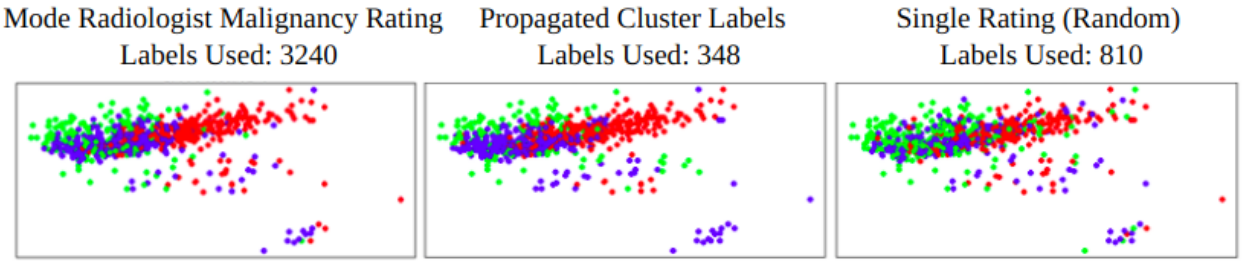

Figure 5. Cluster labels plotted on the space of the first two principal components. The left plot is using the mode of all four ratings (3240 total labels), the center is the propagated cluster labels (348 labels), and a single random label (810 labels).

Figure 6 shows the mode rating compositions of the four biggest clusters. Red is 3, blue is 2, and green is 1. All four case examples have a majority rating, or a rating to which over half of the points in the cluster are assigned. The largest percentage of ratings within a cluster is the purity of a cluster. For example, the rightmost cluster in Figure 6 is considered 64% pure. The average weighted purity across all 87 clusters is 71.6%.

It should be noted that even the mode of all four malignancy ratings is not the ground truth for a sample. Therefore, clusters that are not completely pure in terms of mode ratings may not necessarily be poor clusters. In addition, for some instances the propagated label of a cluster will not be the majority label within a cluster.

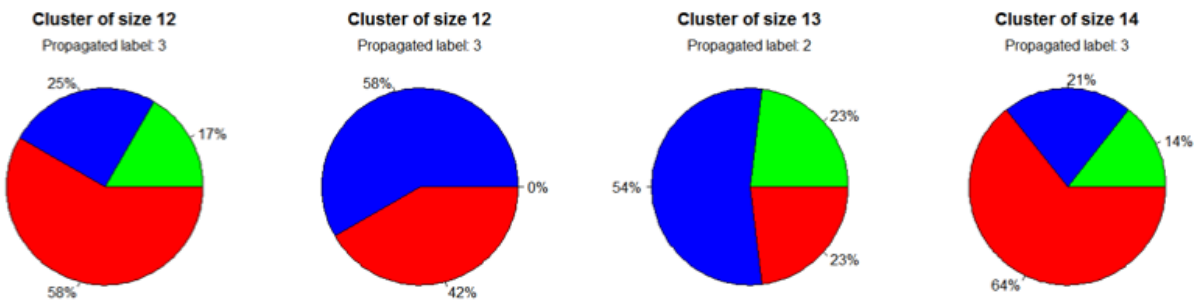

Figure 6. Purities within four largest clusters assigned by mode. Red is class 3, blue is class 2, green is class 1.

In this scenario, we expect two different outcomes: 1) the propagated label was incorrect and a majority of the points in the cluster will request an additional label in the next iteration or 2) the propagated label holds as a strong truth for the points in the cluster and will train a better classifier.

4.2 Classification Performance and Cost

Table 1 summarizes the performance of our method compared to SIC. Since our algorithm was repeated 20 times to control for label order bias, we report averaged accuracy and cumulative label counts for each iteration. Marginal benefit is the increase in accuracy with each iteration.

Approach	Iteration (20 trials each)	Accuracy (%)	Marginal Benefit (%)	Cumulative Label Count
Propagated SIC with Pruning	1	51.35		533.2
	2	71.54	20.19	802.2
	3	78.40	6.86	956.1
	4	82.84	4.44	1064.5
Propagated SIC without Pruning	1	50.12		533.2
	2	68.21	18.09	795.8
	3	75.62	7.71	961.0
	4	79.51	3.89	1087.1
SIC	1	52.87		810
	2	65.14	12.27	1110.4
	3	72.20	6.88	1330.3
	4	77.68	5.66	1502.6
Propagated SIC Using Single Random Label	1	53.03		241.2
	2	68.51	16.48	491.2
	3	75.24	6.73	674.0
	4	79.25	4.01	809.1

Table 1. Iterative performance results on the test set between methods.

We executed a paired t-test, since our datasets were consistent between methods. Our alternative hypothesis is that the true difference in mean accuracies between our method and SIC at iteration four is greater than zero. The test yielded a p-value of 0.01.

As expected, accuracy improves with each iteration. Our method uses 29% fewer labels than in SIC, while achieving a 5.2% higher accuracy. Our methods highest accuracy can be attributed to tree pruning, which was not executed in standard SIC. However, our method still has a higher accuracy without pruning. This gain in accuracy is due to the improved label quality in the first iteration.

Table 1 also shows the result of propagating a single random label from the representative sample of a cluster rather than the mode. This single selection method resulted in a lower accuracy and higher final label count when compared to the mode. The results demonstrate the need to minimize uncertainty for the label to be propagated. By acknowledging closely related points in the feature space and establishing a strong ground truth for the representative of each cluster, we are able to form accurate reference truths that result in better classifiers.

Table 2 shows the sensitivities and specificities for the final propagation selection model. They are measured for each class using the one vs. all approach; the chosen class is considered positive and the other two are negative. Class 2 and 3s specificities exceed their sensitivities, while the converse is true for class 1.

Class	Sensitivity (%)	Specificity (%)
1	97.49	89.93
2	81.30	92.20
3	83.71	94.55

Table 2. Sensitivities and specificities for each class of the final propagation selection model.

4.3 Five-Class Classification Problem

Without label mapping, predicting the malignancy becomes a more difficult, unbalanced 5-class problem. Our propagation and the SIC algorithms were tested without label mapping to test for robustness to unbalanced datasets. The propagated selection algorithm maintained a higher accuracy and lower cost when compared to the SIC, as shown in Table 3.

Approach	Iteration (20 trials each)	Accuracy (%)	Marginal Benefit (%)	Cumulative Label Count
Propagated SIC	1	43.95		432.0
	2	63.95	20.00	728.3
	3	72.59	8.64	913.1
	4	75.86	0.27	1048.3
SIC	1	42.84		810
	2	56.11	13.27	1166.7
	3	64.51	8.40	1435.7
	4	68.89	4.38	1647.0

Table 3. Iterative performance results on the test set between representative labeling strategies on the 5-class classification problem.

Once again, we executed a paired t-test for this 5-class problem. Our alternative hypothesis is that the true difference in mean accuracies between our method and SIC at iteration four is greater than zero. The test yielded a p-value of 0.00.

5. CONCLUSION AND FUTURE WORK

Our analysis demonstrates the benefits of using feature data distribution to guide active learning. Our methods save labels in model training while still improving accuracy and reducing uncertainty in labels.

While our methods were inspired by medical image data, they could be expanded to any scenario with uncertain and expensive expert labels. Future work will apply our algorithm to new datasets to assess robustness. Furthermore, our methods could benefit from more advanced clustering techniques. This would facilitate propagation of accurate labels and could potentially rely on less cluster representatives.

ACKNOWLEDGMENTS

S. Berglin and E. Shin were supported by the National Science Foundation under the Research Experience for Undergraduates grant number 1659836. The authors also acknowledge the NIH National Cancer Institute for its crucial role in the creation of the free publicly available LIDC database used in this study.

REFERENCES

[1] Armato III, S. G., McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., Zhao, B., Aberle, D. R., Henschke, C. I., Hoffman, E. A., et al., "The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans," *Medical physics* **38**(2), 915–931 (2011).

[2] Sheng, V. S., Provost, F., and Ipeirotis, P. G., "Get another label? improving data quality and data mining using multiple, noisy labelers," in [*Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*], 614–622, ACM (2008).

[3] Reiner, B. I., "A crisis in confidence: A combined challenge and opportunity for medical imaging providers," *Journal of the American College of Radiology* **11**(2), 107–108 (2014).

[4] Riely, A., Sablan, K., Xiaotao, T., Furst, J., and Raicu, D., "Reducing annotation cost and uncertainty in computer-aided diagnosis through selective iterative classification," in [*Medical Imaging 2015: Computer-Aided Diagnosis*], **9414**, 94141K, International Society for Optics and Photonics (2015).

[5] Jin, R. and Ghahramani, Z., "Learning with multiple labels," in [*Advances in neural information processing systems*], 921–928 (2003).

[6] Mavandadi, S., Feng, S., Yu, F., Dimitrov, S., Nielsen-Saines, K., Prescott, W. R., and Ozcan, A., "A mathematical framework for combining decisions of multiple experts toward accurate and remote diagnosis of malaria using tele-microscopy," *PloS one* **7**(10), e46192 (2012).

[7] Raykar, V. C., Yu, S., Zhao, L. H., Jerebko, A., Florin, C., Valadez, G. H., Bogoni, L., and Moy, L., "Supervised learning from multiple experts: whom to trust when everyone lies a bit," in [*Proceedings of the 26th Annual international conference on machine learning*], 889–896, ACM (2009).

[8] Yan, Y., Rosales, R., Fung, G., and Dy, J. G., "Active learning from crowds.," in [*ICML*], **11**, 1161–1168 (2011).

[9] Nguyen, H. T. and Smeulders, A., "Active learning using pre-clustering," in [*Proceedings of the twenty-first international conference on Machine learning*], 79, ACM (2004).

[10] Dasgupta, S. and Hsu, D., "Hierarchical sampling for active learning," in [*Proceedings of the 25th international conference on Machine learning*], 208–215, ACM (2008).