TRAINING SET SELECTION TO IMPROVE CROP CLASSIFICATION

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Eric John Christeson

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

December 2014

Fargo, North Dakota

NORTH DAKOTA STATE UNIVERSITY

Graduate School

Title

TRAINING SET SELECTION TO IMPROVE CROP CLASSIFICATION

By

Eric John Christeson

The supervisory committee certifies that this thesis complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Anne Denton
Chair

Dr. Juan Li

Dr. Saeed Salem

Dr. Stephanie Day

Approved:

October 20, 2015
Date

Dr. Brian M. Slator
Department Chair

ABSTRACT

In some classification problems, acquiring class label information is much more expensive than collecting attribute data. One such problem is crop classification from satellite imagery. While random sampling is one option, we demonstrate that a targeted training set selection can be beneficial, when the acquisition of class label information occurs in sets: In crop classification, the number of data points is given by the number of pixels in the imagery, while all data points within one field can be assumed to have the same class label. Each data point is constructed from multiple images throughout the growing season, and each field corresponds to a multi-dimensional distribution of those data points. We demonstrate that it is beneficial to use clustering to select the fields for class label collection. Using this technique, we show that crop classification for partially labeled data can be substantially improved.

ACKNOWLEDGEMENTS

I would like, first and foremost, to thank Stephanie, Ian, and Zach. My wife and kids have been patient with me throughout my graduate career and especially as I've been conducting my research. They put up with my late nights, missing bed-time, and all the times that I was 'off in space'. Without their encouragement, patience, and understanding none of this would have been possible. I'd also like to thank long-time friend Kirk Stueve for planting the seeds for this research. Anne Denton has been a phenomenal advisor. She has been very helpful proof-reading and providing guidance toward new avenues of attack when I was stuck. I would also like to thank the other committe memebers for their time and suggestions.

This material is based upon work supported by the National Science Foundation through grants PFI-1114363 and IIA-1355466. We thank American Crystal Sugar Company for working with us on the sugarbeet portion of the analysis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Remotely sensed data are becoming ever more prevalent and the cost of collecting such data is dropping. Even the publicly available Landsat data provide a complete coverage of the earth every 16 days. In contrast, it is much more expensive to collect some other pieces of information. An example is the type of crop that is grown on an agricultural field. Gathering this information requires that someone physically inspects the field in question. It is possible to query the owners of each field, but this is also labor intensive and results may not be very good; owners may not know what is planted in each field or may be reluctant to share the information. This information is collected at a county level, but is not public information. Another possible solution is to use a computer to analyze and classify fields based on remotely sensed data. There are efforts to classify agricultural land using satellite imagery, but most of these methods suffer from one of two problems. Either they have limited accuracy or take a relatively large effort to classify the training set. We want training samples that are most helpful towards the classification goal. If we need to physically check a field to find out what type of crop is grown there, a time and resource intensive process, we want to pick the fields that do the best job of training. Under these circumstances, it is beneficial to select a small sample of fields for a training set.

Figure 1.1 illustrates a series of raster image with fields. These images are composed of a number of pixels. Each pixel of the image corresponds to a specific area on the ground and, since Landsat images are geo-referenced, we can use a series of these images taken over a period of months to generate a temporal-spectral profile of the Normalized Difference Vegetation Index (NDVI) values. It has been shown that individual crops produce profiles that can be used to identify the crop [14]. The value of each field in an image is calculated by averaging the values of the pixels contained within that field. An example of this is shown in figure 1.2. In our method, the temporal-spectral profile of each point in the field is represented by a vector of NDVI values, one from each image in the series as shown in figure 1.3. In the classification step, each of these vectors is labeled individually but the majority of the field's points determine the label for that field. Since the class labels are collected at the level of a field, one way to select a set of fields for training would be to select fields at random. We show that it is beneficial to use a different method,

ultimately clustering, to select a set of fields for training a classifier.

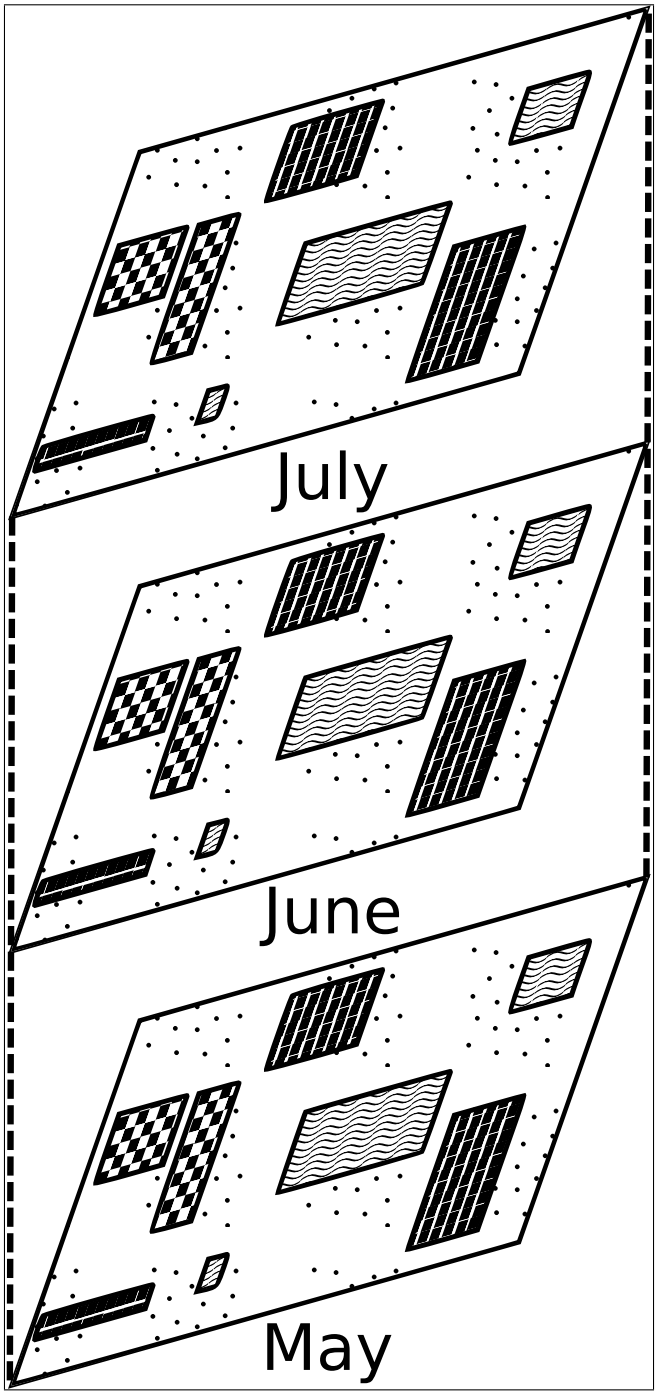

Figure 1.1. Schematic representation of a satellite image time series showing fields with different crops. The images are georeferenced so that a pixel in one position on an image represents the same location on earth in the other images. This allows the data from each specific pixel location to be used as a time series vector with each image contributing a value for its corresponding time.

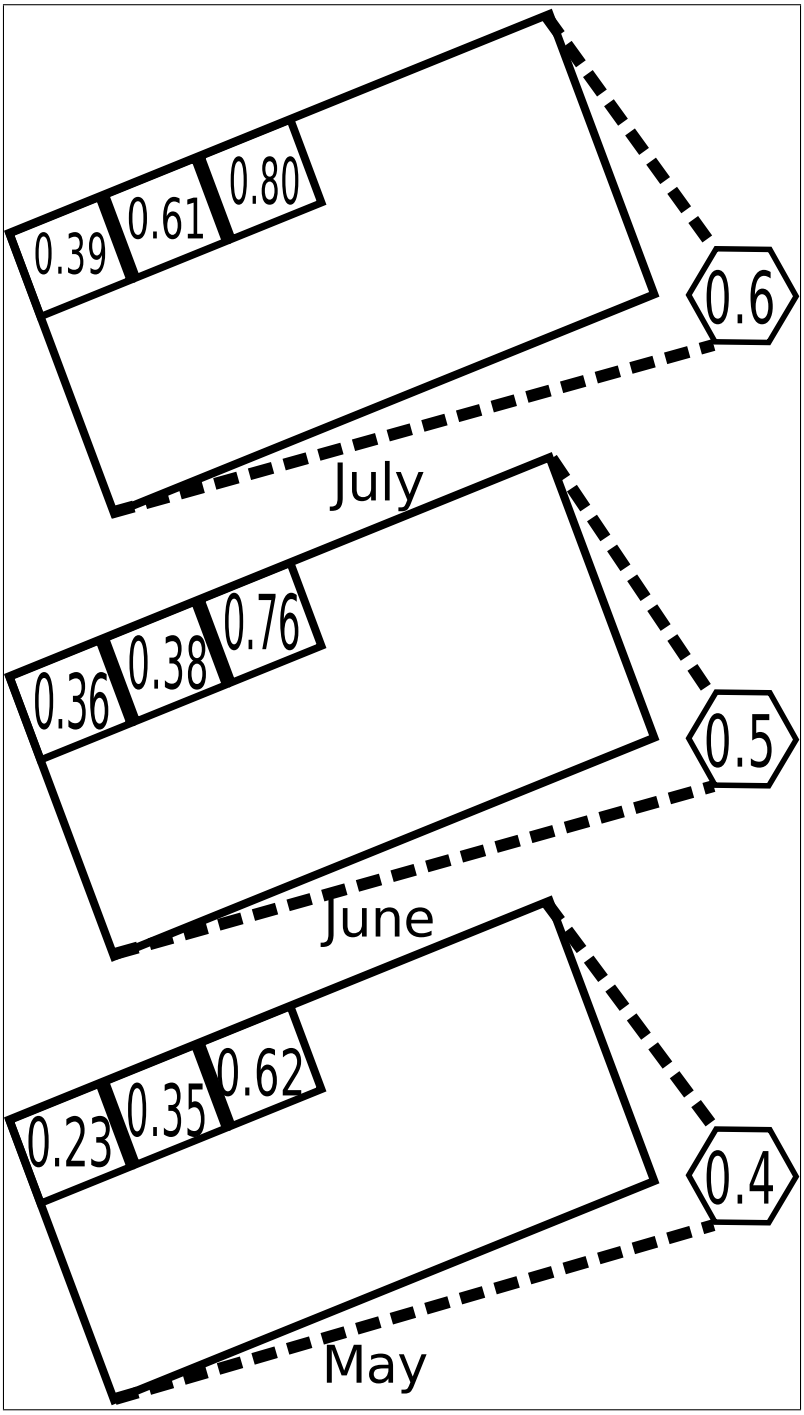

Figure 1.2. This is a representation of satellite images for one field from May - July. The NDVI values from each pixel in the field are averaged into a single value for the whole field. This average value is used in classification instead of the value of each pixel. In this case, the values shown from May are averaged to a value of 0.4 for May. June and July are treated similarily resulting in values of 0.5 and 0.6, respectively. This allows classification at the level of a field and significantly reduces the number of entities that must be classified as each field may have hundreds or thousands of pixels.

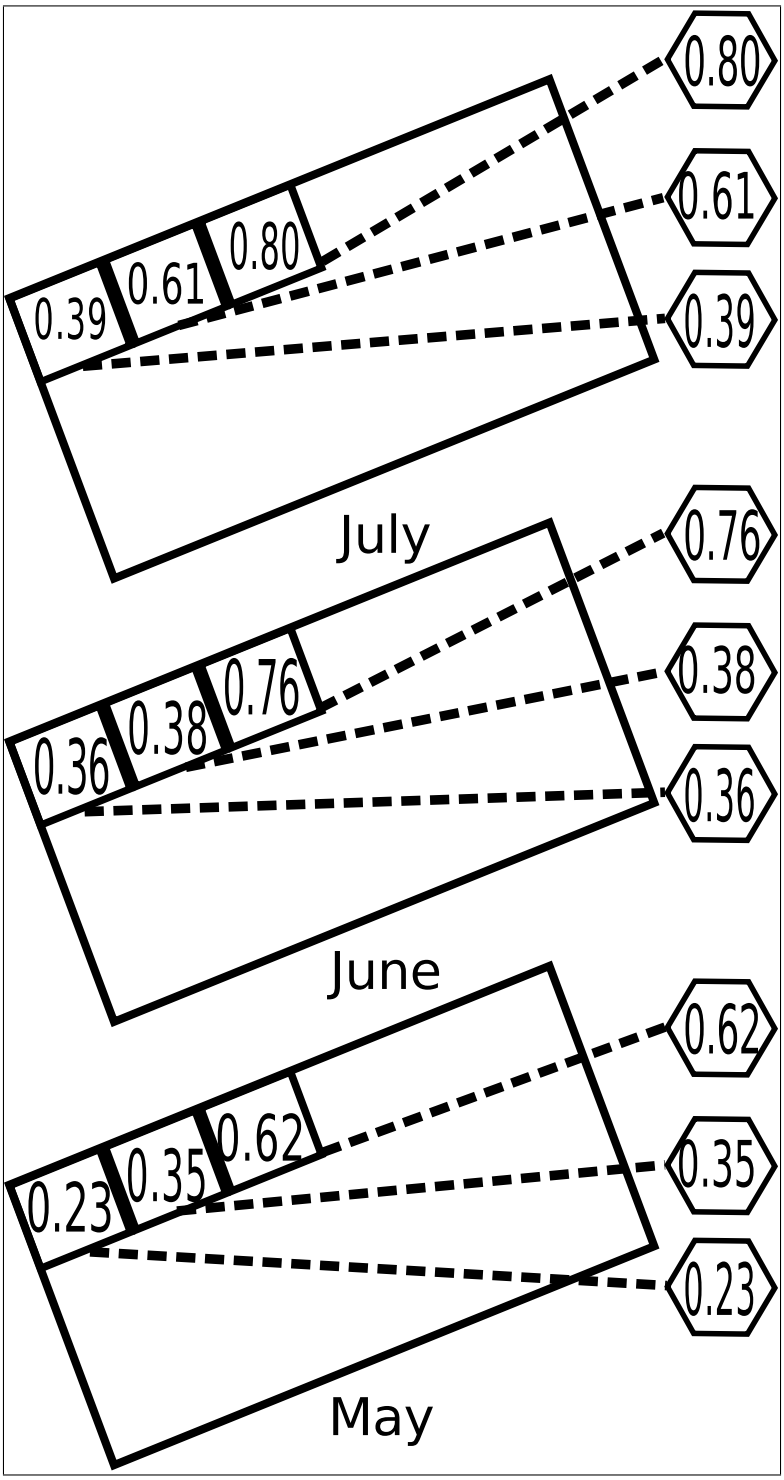

Figure 1.3. The same field from May - July. The NDVI values from each pixel are used to classify pixels individually rather than classifying the field as a whole with an aggregate value. These individual classifications are then used to classify the field as a whole as explained in Section 2.4. This increases the number of classifications, but results in better classification accuracy.

1.1. Problem Statement

Impact: Knowing what crops are grown over a region has far reaching impact for other researchers, government officials, and people who just want to know. The effort required to collect this information manually is large, and even though it is currently collected at a county level, this approach could be used to verify the collected data.

Problem: Using the mean value at the field level before training and classification loses information. This happens because of the variability in plant health and thus NDVI values across a field due to water, soil conditions, or other factors. Keeping this information could improve classification results.

Concept: Classification will be based on a time-series of satellite images from a growing season, typically April to September. A data set is available that includes field boundaries and whether or not sugar beets were planted in the field for a particular year. We will show that performing aggregation of pixels after labeling instead of aggregating before training is beneficial. We will then explore methods of selecting a small set of fields to train a classifier. The goal will be to find a selection method that uses the fewest fields for training but still produces better results than selecting a random training set.

1.2. Background

Previous work on reducing the cost of assigning class labels to the training set has focussed on semi-supervised clustering [1, 3]. Other work [4] proposes a way to measure and minimize the cost of selecting a training set. Our approach guides training set selection through clustering. Instead of working with unlabeled data, we make use of the large number of pixels that come from only a few labeled fields that are selected through clustering.

2. CONCEPTS

As mentioned in the problem statement (see Section 1.1), using a mean or other aggregate value for training and classification causes information to be lost. In order to test this, we considered two different methods of preparing the data. They are mean and each cell which are explained in Sections 2.3 and 2.4 respectively. Once the data preparation method is chosen, we must select a set of fields for training our classifier. We considered four methods; one baseline and three test methods. Each of these were combined with the mean and each cell preparation methods to produce a total of eight different methods. The baseline method, random, is described in Section 2.5. As we worked toward meaningful results, we considered two methods of selecting a training set, K-L divergence (see Section 2.6) and information gain (see Section 2.7. These methods did not produce favorable results, but, for the sake of completeness, we will discuss them. The final method we will present is clustered (see Section 2.8). This method did produce meaningful results which are discussed in depth in Results (see Section 4).

2.1. Classification

In classification we seek a determination as to which of a set of categories a new observation belongs. A common example is classifying an e-mail message as "spam" or "not-spam". A set of observations with known category membership, called a training set, is used set up, or train, a classifier to label observations with unknown category membership. Since it relies on a set of correctly labeled observations, classification is an instance of supervised learning. This contrasts with clustering which does not depend on labeled observations and as such is an unsupervised learning method. One liability of classification is that the training set is sensitive to underrepresentation of classes. In particular if one has a training set with only positive labels, the classifier will not be able to predict a negative label. We ran into this problem with the information gain method (see Section 2.7) as described later. The classifier we used is based on the commonly used C4.5 [15] algorithm. We use the default settings in the Weka [7] J48 classifier.

2.2. NDVI

We will use a calculated value called the Normalized Difference Vegetation Index (NDVI). This value is calculated from the spectral reflectance in the visible red and near-infrared bands.

These reflectance values are ratios of reflected over incoming radiation in each spectral band. The formula is

$$NDVI = \frac{(NIR - VIS)}{(NIR + VIS)}$$

so the values for NDVI range from -1.0 to 1.0. It is commonly used to classify land cover [17, 10].

2.3. Mean

The mean method uses the mean NDVI value of the data points within each field's boundaries in training and classification. The mean value is calculated separately for each satellite image, giving a vector of mean values for each field. This vector is then used for training and classification. This method reduces the number of vectors to one per field because of the aggregation. This reduces the time needed for training and classification.

2.4. Each Cell

In the each cell method, each data point vector is considered individually for both the training and classification steps. This often results in point that belong to the same field being classified differently. Since the focus is in classifying fields, a field's classification is an aggregation of the classifications of its set of points. While any number of different aggregation methods may be used, we chose to label a field according to the majority of its points. The difference between this method and the mean method is when the aggregation occurs. Here, the aggregation happens after the classification compared with before classification for the mean method.

2.5. Random

Since randomly selecting a set of fields for a training set is trivial to implement, we used it as a baseline for our comparisons. We refer to this method as random.

2.6. Kullback-Leibler Divergence

This method uses Kullback-Leibler divergence [13] to select a training set. The K-L divergence is defined as $D(p_1 \| \, p_2) = \int p_1 \log(p_1/p_2) \mathrm{d}\lambda$. In short, this is a measure of difference between two distributions. We first compute the K-L divergence between a field and the rest of the fields, then order the fields according to this divergence. We tried selecting the fields with either the highest or lowest K-L divergence for a training set.

2.7. Information Gain

Similar the the K-L divergence, in this method, we compute the information gain between a field and the other fields. The fields are ordered by the information gain value and fields with either the highest or lowest information gain are selected for training.

2.8. Clustered

Clustering is partitioning a set of data into subsets such that items in the subset, or cluster, are more similar to each other than to items in other clusters. It is a form of unsupervised learning as the data is unlabeled. We used a centroid-based clustering known as k-means in which we specify the number of clusters, k, ahead of time. It is an optimization problem that seeks to find k cluster centers and assign the items to the nearest center, minimizing the sum of squared distances to the cluster center. In our process we treat the point vectors as points in n-dimensional space and use the Hartigan and Wong algorithm [9] as implemented in the R [16] core, to partition the point vectors into k clusters. Fields are chosen in multiples of k so N fields from each cluster are chosen for training. We chose the fields within each cluster randomly, but we hope to explore other methods of choosing fields from the clusters.

2.9. Algorithms

Figure 2.1 shows the algorithm used to clean up the data. The missing value thresholds of 50% were found to remove much of the missing data without removing too many rows and thus removing too much good data with it. Since most of the fields have hundreds or even thousands of pixels and thus rows of data, removing those with fewer than 50 rows was deemed prudent to keep field size more balanced.

The algorithm we use for selecting a training set is shown in Figure 2.2. Most algorithms for k-means give an approximate solution since the optimization problem is NP-hard. An approximation works fine in our case since we cluster by point vector, but assign each field to a cluster based on which cluster has the most point vectors from that field. We choose the number of fields in our training set to be a multiple of the number clusters we have. We can choose our training set from the clusters by picking a random field from clusters 1 to k until we have the desired number of fields. Since the number of fields is a multiple of the number of clusters, each cluster will contribute the same number of fields to the training set. Selection of the training set is performed at the field

level, but the training set is the set of point vectors belonging to those fields. This means that the number of fields used for classification is predefined, but the number of data points used for classification depends on how many data points those fields contain.

Columns of data: C
Rows of data: R

1. Remove columns with $> 50\%$ of values missing
2. Remove rows with $> 50\%$ of values missing
3. Remove rows with either first or last values missing
4. Remove fields with < 50 rows
5. **for each** row $r \in R$ {
6. replace missing values using
7. linear model $x + x^2 + x^3$
8. }

Figure 2.1. Algorithm for smoothing missing values

Set of candidate fields for training: I
Set of pixels: I^s
Set of clusters: K
Number of clusters: N
Vector of values for a pixel: v
Desired number of fields for training: t
Training set: S

1. Use k-means to cluster all pixels $p \in I^s$
2. **for each** field $f \in I$ {
3. **for each** cluster $K_n \in K$ {
4. count pixels from $f \in K_n$
5. }
6. assign f to cluster K_n with the most pixels from f
7. }
8. **for each** cluster $K_n \in K$ {
9. Select t/N fields from K_n and assign to S
10. }

Figure 2.2. Training Set Selection Algorithm

3. IMPLEMENTATION

The process we use to compare methods is split into three major steps as shown in Figure 3.1. The first step is selecting either mean or each cell method to prepare the data. Next, we choose one of the training set selection methods: random, K-L divergence, information gain, or clustering. The selected training set is used to train a classifier. Finally the testing set is classified and the accuracy of the classification is calculated against the known classification. This was repeated for each combination of preparation and training set selection and the accuracy of each method was compared.

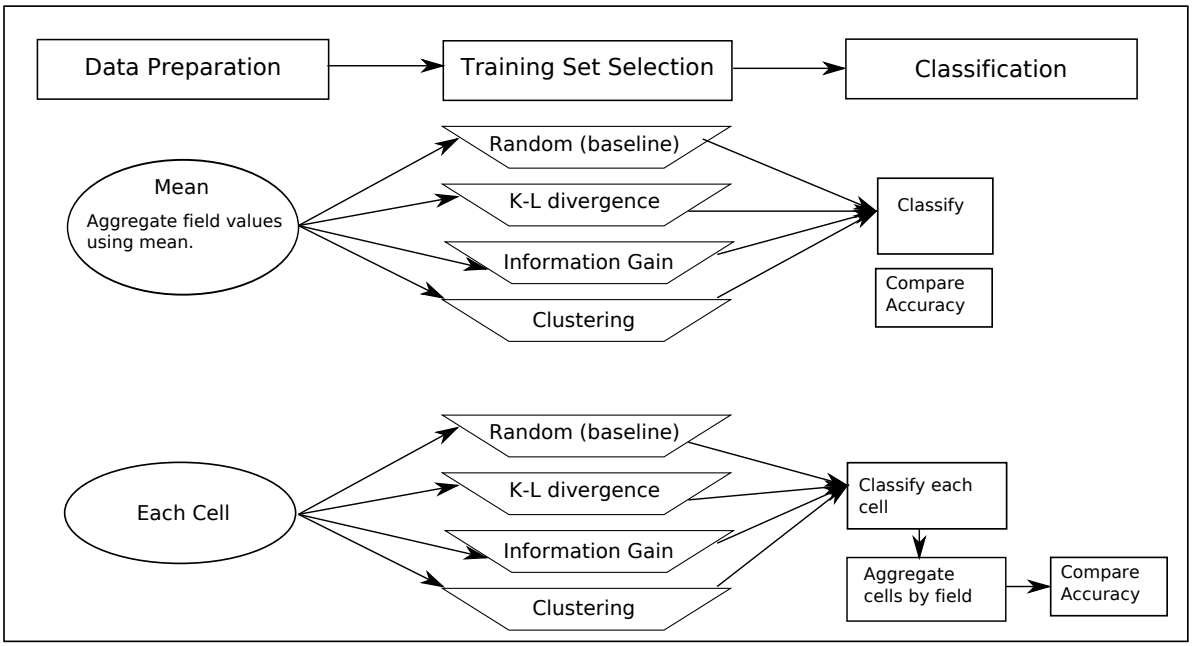

Figure 3.1. Schematic overview of the process shows the three general steps used. First, the data is prepared either by leaving each vector of pixel data as is, or by using the mean value for each field. Second, a training set is selected using one of the four selection methods. This training set is used to train a classifier and the test set is classified. After this, the classification accuracy for each method is compared.

3.1. Data

The data consists of 6 years of field boundary data (2007 - 2012) and 5 years (2007 - 2011) of Landsat 5 images. The field boundaries are in ESRI's shapefile format. Each field has auxiliary

Table 3.1. Number of fields in each set

Year	Positive fields	Negative fields
2007-2008	4118	3407
2008-2009	3707	3945
2009-2010	3267	3181
2010-2011	3438	3717
2011-2012	3895	3676

data such as the coordinates of its centroid. The set of fields in a particular year's file were all planted with the same crop and as such, were used as our positive class for training and evaluating. For a negative set, we used the next year's fields. Standard farming practice precludes planting the same crop in a field in adjacent years and a survey of the data supported this assumption. We use two years to define a set of positive samples and negative samples (e.g. 2007-2008). Choosing the data in this manner helped balance the positive and negative items. Table 3.1 shows the number of fields used for each year. The boundaries were cropped by 2 image pixels (60m) to mitigate any inaccuracy in field boundaries and lessen effects from adjacent fields or roads. The Landsat 5 rasters were first processed to remove areas of cloud cover using the Automated Cloud-Cover Assessment algorithm [11, 12] as implemented in GRASS GIS [6]. The we calculated a Normalized Difference Vegetation Index (NDVI) value for each pixel. This value is used as an indicator of biomass and to differentiate vegetation from water or bare soil. The NDVI varies over a growing season, and is commonly used for classifying land cover. NDVI is a continuous value which varies from -1.0 to 1.0. In order to easily export the data from GRASS and convert it for use in later stages, the NDVI values were scaled from 0 to 20000 by adding 1, then multiplying by 10000. This gave us a number that would fit into an int16 data type which resulted in a file 1/2 to 1/4 the size of exporting to float32 or float64. We used the GeoTrellis framework [5] to perform zonal operations and generate a CSV file. A zonal operation applies a mathematical operation (for example min, max, mean, addition of a constant) to the values of a raster's pixels which are contained within a polygon, or zone. The first set of files generated were positive attribute, mean-values. The shapefile for positive attribute for a year was loaded and a set of polygon objects representing each field was generated. Fields were identified by calculating the coordinates of the polygon centroid. A set of rasters images with NDVI values for each pixel was loaded representing each Landsat image collected during the

Table 3.2. Dates of Landsat 5 raster images used for each year

Year	2007	2008	2009	2010	2011
Dates	04-14	05-18	05-27	04-06	04-25
	05-16	06-19	07-08	04-22	06-28
	07-19	07-21	08-25	05-08	07-30
	09-21	08-06	09-10	07-11	10-02
			09-26	09-13	
				09-29	

April through September time frame for that year. A zonal mean operation was run for each polygon on each raster resulting in a single number for each field - raster combination. The values were aggregated on field centroid with a vector of values one for each raster. These were then output to a CSV file along with an identifier of 'positive' to indicate they were the positive values. One row was output for each field. This process was repeated for the negative attribute using the shapefile for the following year to generate the polygons, and the raster from the target year. As already discussed, these fields would not have the target crop planted on them for this year. This produced a CSV file similar to the positive file but with 'negative' in the attribute column. The positive and negative CSV files for a year were concatenated to form the classifier input file for that year. For each of the three methods depending on each cell data, a similar process was followed. First we loaded a shapefile for the year of interest and generated polygon objects for the fields. Then we loaded the set of rasters for that year and read the NDVI value for each pixel in each field. The pixels were grouped according to field, using the fields centroid as an identifier. Once the set of rasters were processed, the values were output as a series of rows, one per pixel, with the latitude and longitude set to the centroid of the field the pixel belongs to. This allows us to use each pixel in classification, but keep its membership in a particular field. This process was repeated for the negative values by again using the field boundaries for the following year combined with the rasters from the target year.

The Landsat images had varying amounts of missing values due to cloud cover. We used an R [16] script to remove columns (images) with more than 50% of values missing. This threshold was chosen as it removed columns with a large amount of missing data, while still leaving some data for us to work with. The final number of columns used was between 4 and 6 for each year as shown in Table 3.2. The available images for each year numbered between 11 and 12 – this shows

a large amount of missing data due to cloud cover. Also note that 2008 has 4 dates right in the middle of the season. This leads to rather poor overall results for 2008 as shown in Figure 4.5.

3.2. Splitting The Data

We decided to partition the data into 5 different pairs of testing and training sets. Our approach is similar to 5-fold cross validation, but used the smaller sets for training and the larger sets for testing. Usually in n-fold cross validation, the data is split into n partitions and each partition is used individually for testing with the rest used for training and the n results are averaged. We felt that since we were not testing the classification algorithm, the relative accuracy of the training sets, swapping the normal testing and training sets was the correct thing to do. This approach gives each item the chance to be used in the training set. This also gave us a much larger amount of data to classify for each classification round. This step was only necessary because we wanted to validate the results. In practice, all of the items would be run through the clustering step and a training set selected from all the possible items.

3.3. Classification Process

The KNIME [2] software package was used to automate the classification process. After a wrong turn with the classification algorithm, we decided to use a decision tree classifier as it was relatively fast and worked with the floating point numbers that resulted from the linear model smoothing step. The mean (see Section 2.3) and each cell (see Section 2.4) methods were each combined with random selection to select the training set. We ran the classification 5 times for each number of training fields and averaged the results to come up with a final accuracy number. Since the each cell method actually classified each cell, a custom script was written to compute the classification of each field, and to generate the confusion matrix based on the actual class label of each field. We used Fisher's exact test to compute the p values for each confusion matrix. In all cases it was below 0.05; one confusion matrix out of 2000 had a p value of 0.03. A p value of < 0.05 is commonly taken to indicate that the results are significant. Thus we have high confidence that each confusion matrix is significant. Then we used the confusion matrices to calculate and plot the accuracy for each year.

4. RESULTS

In reaching our conclusions, there were a few things we tried that did not work. This section will explain some of those techniques and our analysis of why they did not work. The process we followed can be broken down into three main areas; data preparation, candidate selection, and classification. Most of the techniques we tried were in the area of candidate selection, though one change was made to the classification area as a result of candidate selection changes. First we will discuss the field selection techniques we tried will discuss the classification changes we made.

4.1. Kullback-Leibler Divergence Results

The thought was that fields with a NDVI distribution that differed the most from the overall distribution would be good candidates for training. In a manner that was similar to the approach already described, we attempted to compare this with a random training set and a training set made from the least divergent fields. Our first results looked promising, a training set of the fields with the largest K-L divergence seemed to beat the accuracy of the other methods by a few percentage-point margin. Figure 4.1 shows a very good example of this. The first problem was noticed when trying to come up with a mathematical explanation for the results. It was noticed that the data had not been properly split into training and testing sets as described in Section 3.2.

Our explanation for these results is as follows. The fields that were the most different from the rest (largest K-L divergence) were used for training and the rest of the fields were used for testing. The result is that these fields were removed from the testing set when they were used for training. The most obvious problem is that the data was not partitioned into testing and training sets. Besides not being a valid way to test the validity of our training set selection method, this won't work in practice as the goal is to classify all the fields. Deciding not to classify some fields would not be appropriate. Once the data was properly partitioned, the large K-L divergence training set performed much worse than a training set of random fields. An example of this is illustrated in Figure 4.2. Because of this, we decided to abandon K-L divergence and try another method for selecting a training set.

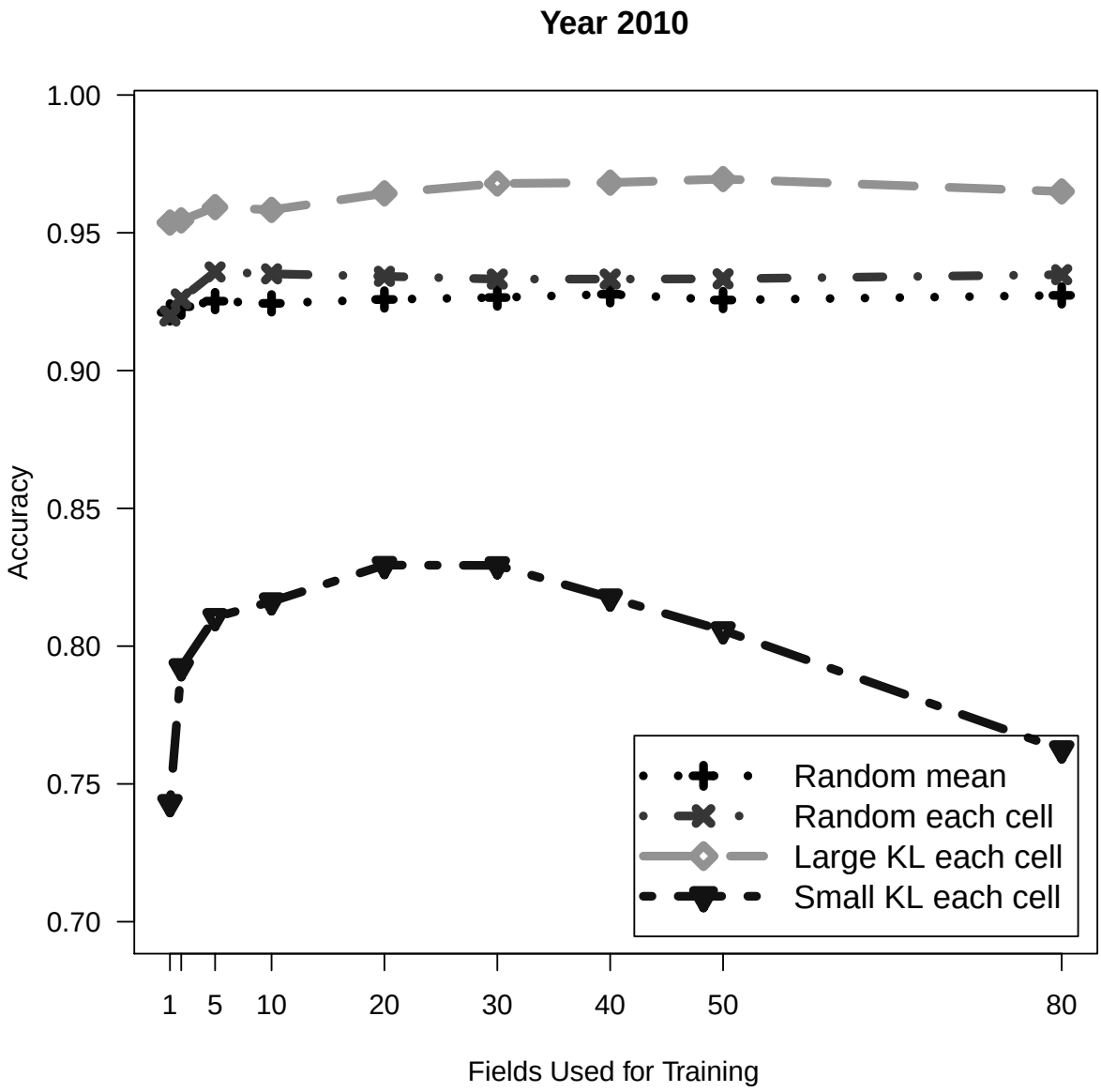

Figure 4.1. Example of the outstanding results we first saw before realizing that the data hadn't been properly partitioned into training and testing sets. These results are for year 2010. The lines show which criteria were used to select a training set. Using fields with the largest K-L divergence (Large KL each cell) show the best accuracy regardless of the percentage of fields used for the training set.

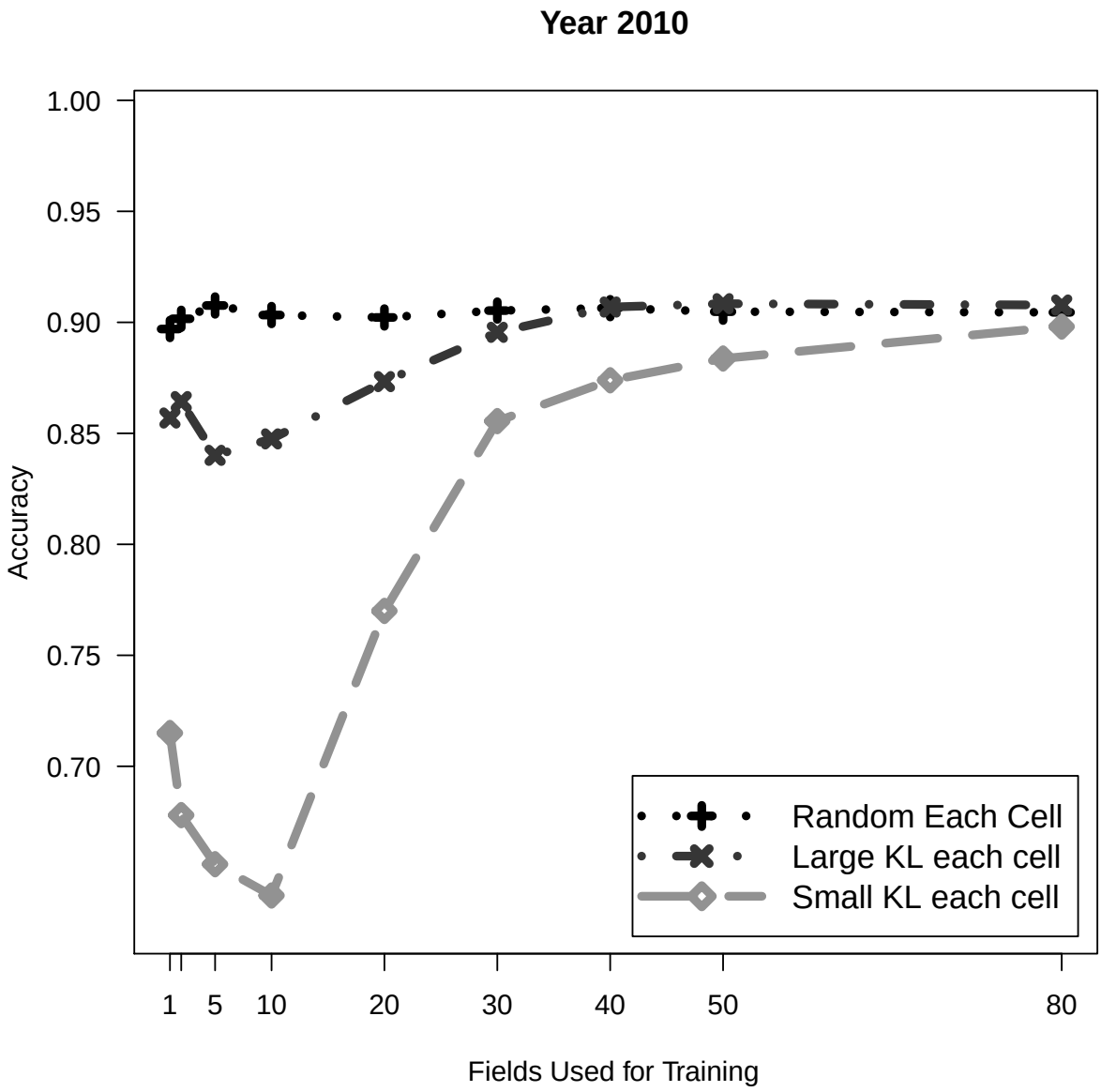

Figure 4.2. Example of change in accuracy after partitioning data into training and testing sets. These results are for year 2010. Once the data was correctly partitioned into testing and training sets, selecting a random set of fields for training (Random each cell) produced a classifier with the best accuracy. Using fields with large K-L divergence doesn't match the random set until about 40% of the fields are used for training. This is an unacceptably high percentage.

4.2. Information Gain Results

The next thing we tried was using information gain to select a training set. Our hope was that a training set with either the highest or lowest information gain would be beneficial for finding a good training set. The results were frustrating. Using 1 percent of the fields for training would result in high or low information gain having much worse accuracy than a random training set. Using 5 - 10 percent of fields would improve the accuracy to near that of the random set. We investigated these results and found that often the 1 percent of fields with the highest or lowest information gain would all have either a positive or negative class label. This would result in a bad training set. If all or almost all of the training set has the same label, the classifier is not trained very well. Each label needs to have some representation in a training set in order to have a well-trained classifier. This realization led to our idea for clustering the fields and choosing from the clusters equally in order to get a balance of labels for training. This has the added benefit of needing to label fewer fields than taking a percentage of the total fields.

4.3. Classification Changes

In practice, our technique for candidate selection should produce good result regardless of the classification algorithm used, however, the classification algorithm should be well-matched to the type of data being classified. Initially we used a Naïve Bayes classifier because of its speed and the fact that it could handle missing values in the NDVI data. We scaled the NDVI data to integer values to work with this classifier. After deciding to use a linear model (see Listing 2.1) to smooth the data and fill in missing values, our classifier no longer needed to handle missing values, and furthermore continuous data requires further processing to use Naïve Bayes [8]. As a result, we decided to use a decision tree classifier. This improved the accuracy for all of the methods.

4.4. Results of Clustered Method

In the clustered method, we cluster each of the point vectors using k-means clustering. We use a k of 10 because it is large enough to cover the most common crops twice, but not so large that the number of fields per cluster gets small. Since each point vector is clustered, it is possible, and in practice likely that a particular field will have point vectors in more than one cluster. In order to ensure that a field is not chosen more than once, each field is assigned to the cluster where most of its point vectors reside.

Figure 4.3 shows a sample distribution of fields after they have been clustered. In our experiment, we looked at using a small number of fields for a training set. When working with K-L divergence and information gain, we focused on using a percentage of the candidate training fields so the number varied by year as there were a different number of fields each year. We used percentages of 1,2,5,10,20,30,40,50 and 80 so, at the 1% level, the number of fields that needed to be labeled for training was 13 to 15. The clustered method used even fewer fields at the lowest level. We evaluated using 10, 20, 60, and 130 fields for training from a total training pool of 1290 to 1531 candidate fields.

When using only 10 fields for training, the results are much worse; the 2011 clustered mean lags the clustered each cell by almost 6.5 percentage points. The random methods tell the same story; in 2007 random mean lags random each cell by 14 percentage point. The reason for this is the larger number of data points used for training when classifying each cell. Since we do not use fields with fewer than 50 pixels, even using 10 fields for training gives us at least 500 vectors to use for training and possibly as many as 20,000.

The second thing we see is that as the number of fields used for training is increased, the accuracy of each method increases. This is expected as we are using with a very low percentage of the candidate training fields for training. The percentage varies from less than 1% when using only 10 fields to around 10% when using 130 fields. In 2009 (see Figure 4.6) the two each cell methods are closest, falling within 0.38 percentage points of each other with the clustered method edging out the random method.

One final thing to note is that the clustered method has a more profound effect when using mean NDVI values. When using 10 fields for training, clustered outperforms random by a minimum of 2.7 percentage points in 2010 and a maximum of 11.5 percentage points in 2007. The absolute results obtained are never as high as when using each cell methods, even with random each cell, but they are much higher than random mean.

The table of results (Table 4.1) and accompanying plots show that using the clustered (see Section 2.8) method to select a set of fields for training outperforms the other methods for selecting training sets. Figure 4.4 shows the results obtained from the 2007 growing season data. There are a few things to note here. First is that our method of classifying individual pixels and aggregating the results clearly outperforms using the field's mean value. This is true whether we use the clustered

method for selecting a training set or not. In fact, even when using 130 fields, using the mean values for classification still lagged using each cell by at least 1 percentage point and by almost 3 percentage points in the worst cases.

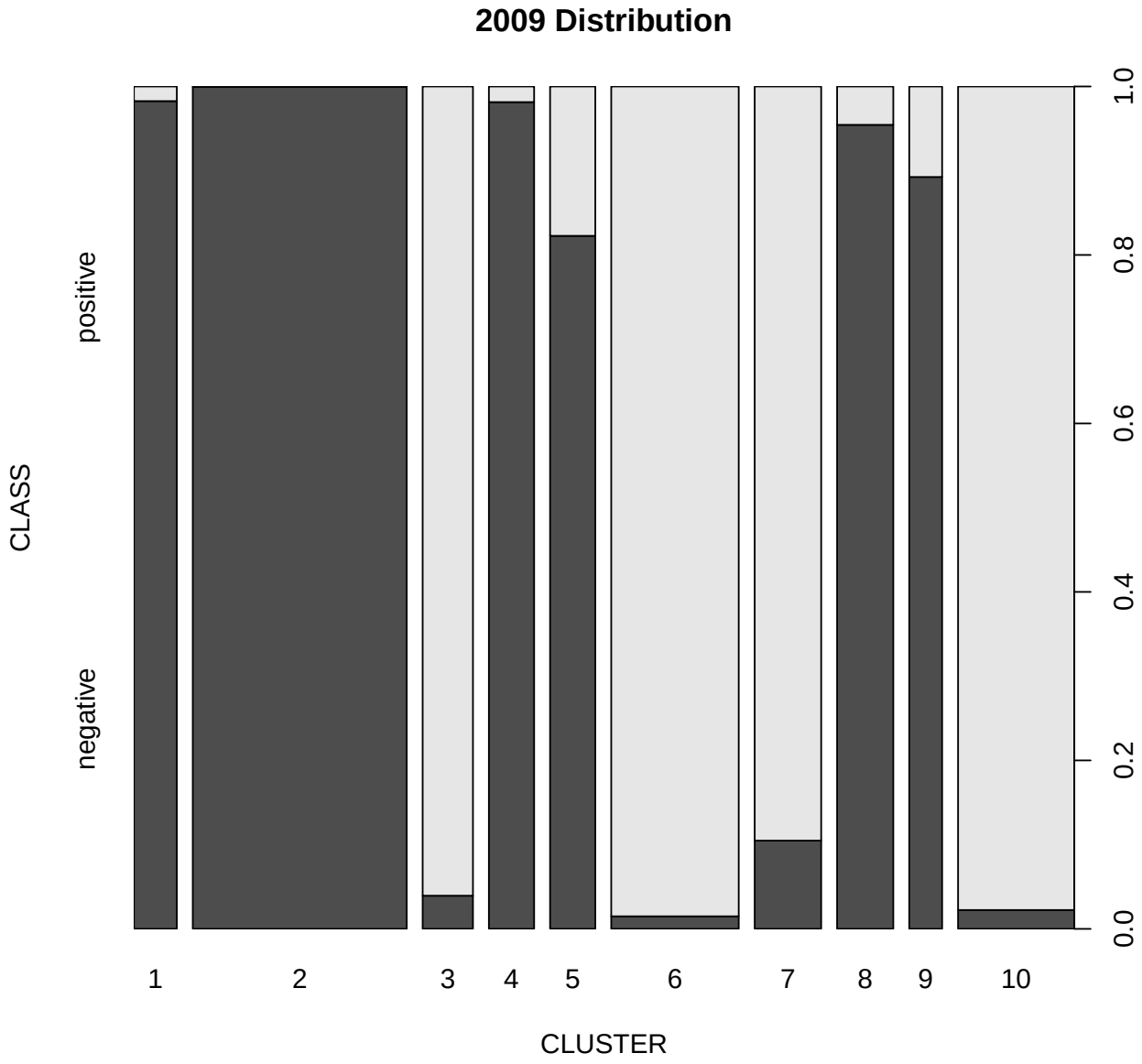

Figure 4.3. Sample Distribution of Clustered Items. Clustering the data provides a good segregation by class value. Using a training set made up of n items from each cluster gives a good probability of a well-balanced training set.

Table 4.1. Table of results. Highest accuracy values in each group are in bold.

Year	Type	Number of Fields used for training			
		10	20	60	130
2007	Clustered, mean	0.8875	0.8984	0.9246	0.9322
	Random, mean	0.7718	0.8899	0.915	0.9294
	Random, each cell	0.9185	0.909	**0.9403**	0.9467
	Clustered, each cell	**0.9228**	**0.9284**	0.9399	**0.947**
2008	Clustered, mean	0.8116	0.8468	0.8815	0.8859
	Random, mean	0.7439	0.8203	0.8767	0.8857
	Random, each cell	0.8567	0.8802	**0.9036**	**0.9141**
	Clustered, each cell	**0.877**	**0.882**	0.9026	0.9137
2009	Clustered, mean	0.8907	0.9183	0.9398	0.9465
	Random, mean	0.8342	0.9041	0.936	0.9399
	Random, each cell	0.9253	0.9193	0.9445	0.9562
	Clustered, each cell	**0.9291**	**0.9403**	**0.9569**	**0.9641**
2010	Clustered, mean	0.8455	0.8946	0.9271	0.9295
	Random, mean	0.8185	0.875	0.9282	0.9347
	Random, each cell	0.8764	0.8978	0.932	0.9434
	Clustered, each cell	**0.8955**	**0.9123**	**0.9392**	**0.9491**
2011	Clustered, mean	0.8095	0.8608	0.8879	0.9035
	Random, mean	0.776	0.8557	0.8858	0.9052
	Random, each cell	0.8637	0.8774	0.906	0.9223
	Clustered, each cell	**0.8743**	**0.8835**	**0.9139**	**0.9281**

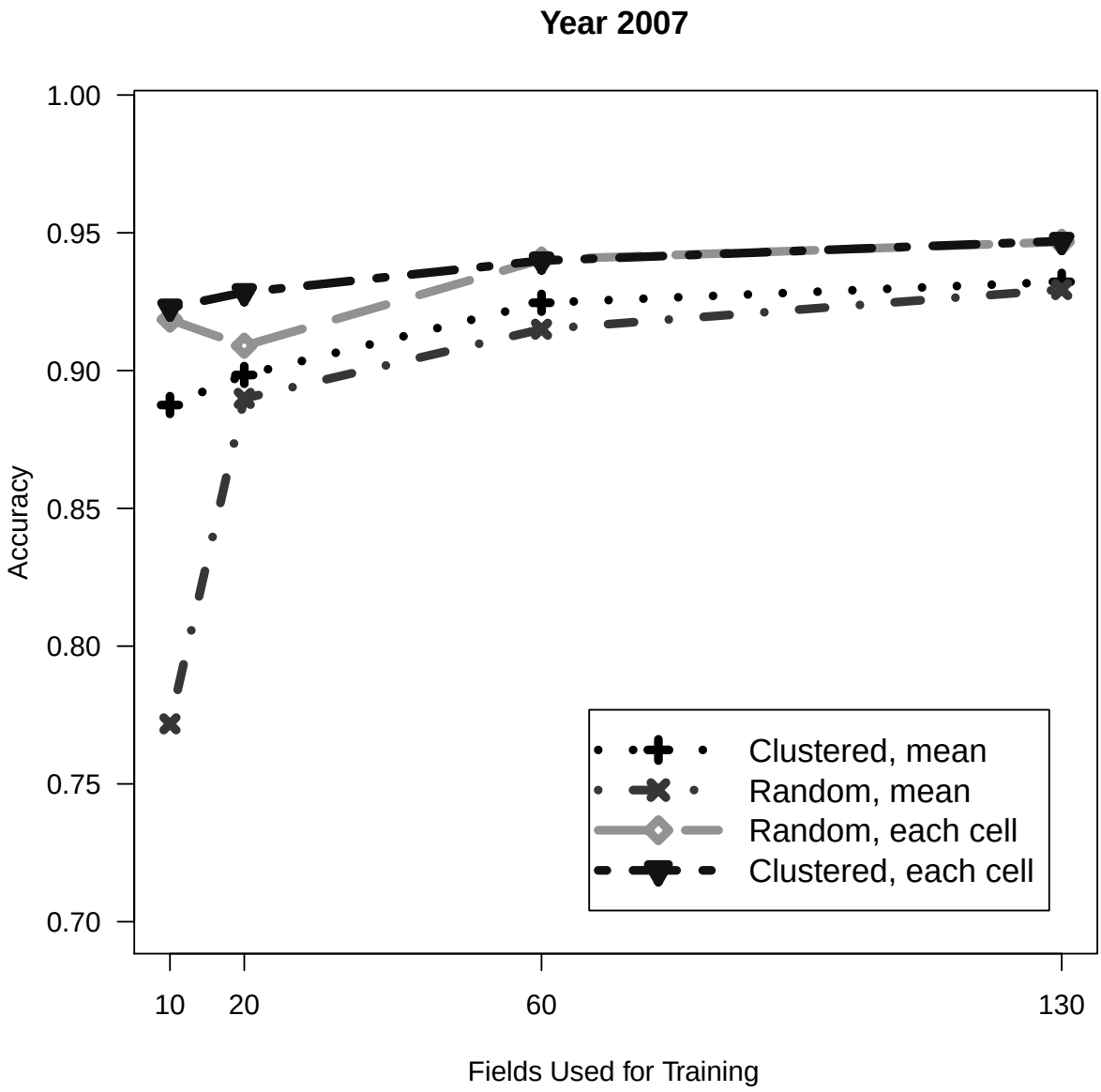

Figure 4.4. The results for year 2007. This plot compares prediction accuracy between random (see Section 2.5) and clustered (see Section 2.8) methods of training set selection and mean (see Section 2.3) and each cell (see Section 2.4) methods. These results show that using each cell results in consistently better accuracy than using the mean for each field. The clustered method shows better accuracy than the random method for low numbers of fields.

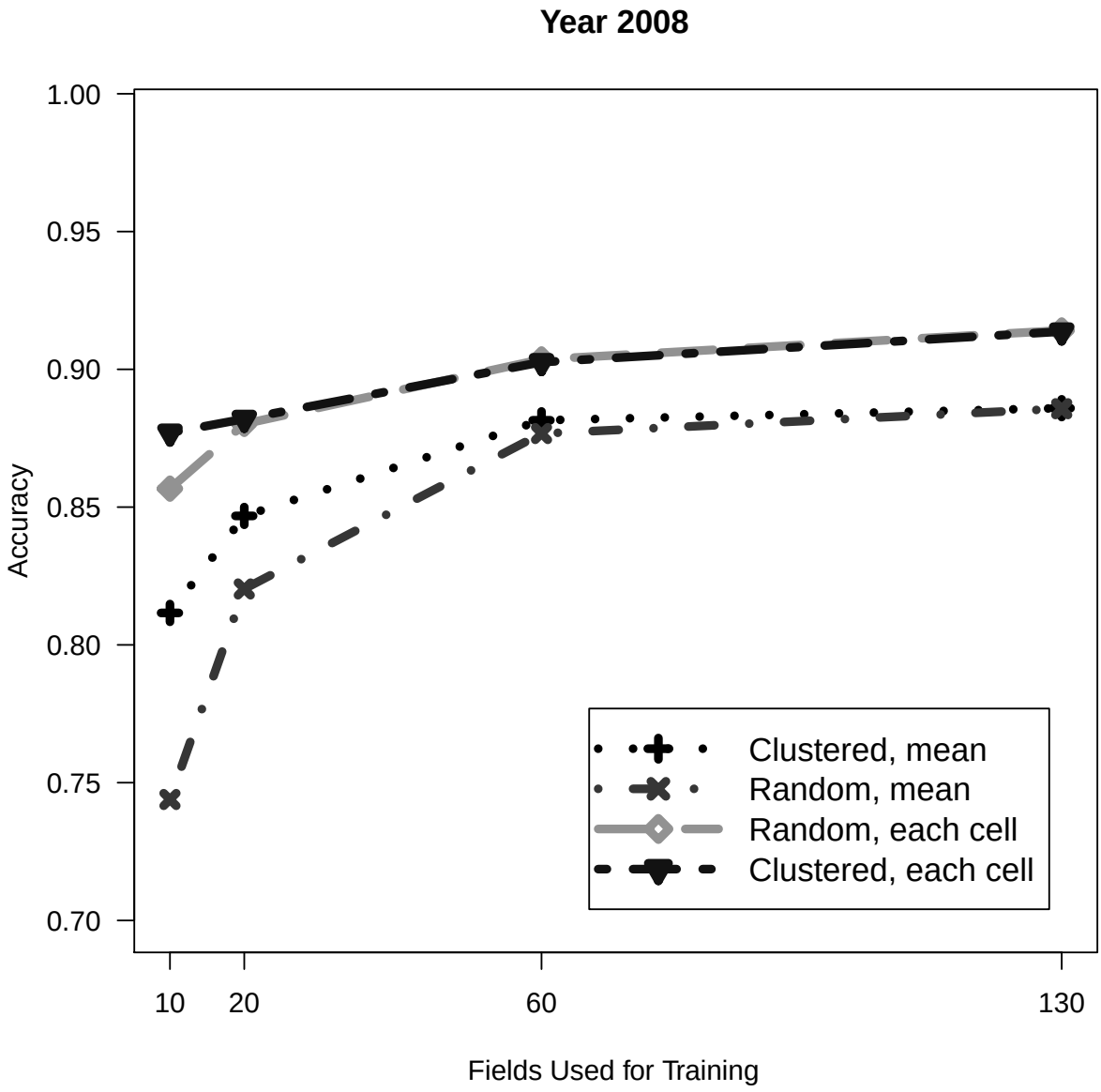

Figure 4.5. These results from 2008 show it edging out 2011 for the worst performing year. Although the highest accuracy with 10 fields is 0.1 percentage points better than 2011, it falls behind across the board as the number of fields used increases. Things which could affect this are an inbalance in the number of positive or negative fields in the pool, or the data itself. An imablance does not seem to be the cause as 2010 has a similar mixture of positive and negative fields (238 more positive fields in 2008 and 279 more positive fields in 2010), yet has overall better accuracy. 2008 has a narrow range of data compared to the other years. It has only 4 images (as do 2007 and 2011) and the second latest starting image, 9 days earlier than 2009 and 23 days later than 2011 which is next. It also has the earliest ending image by 46 days. Crucially, this ending is early in August with at least a month of growing left.

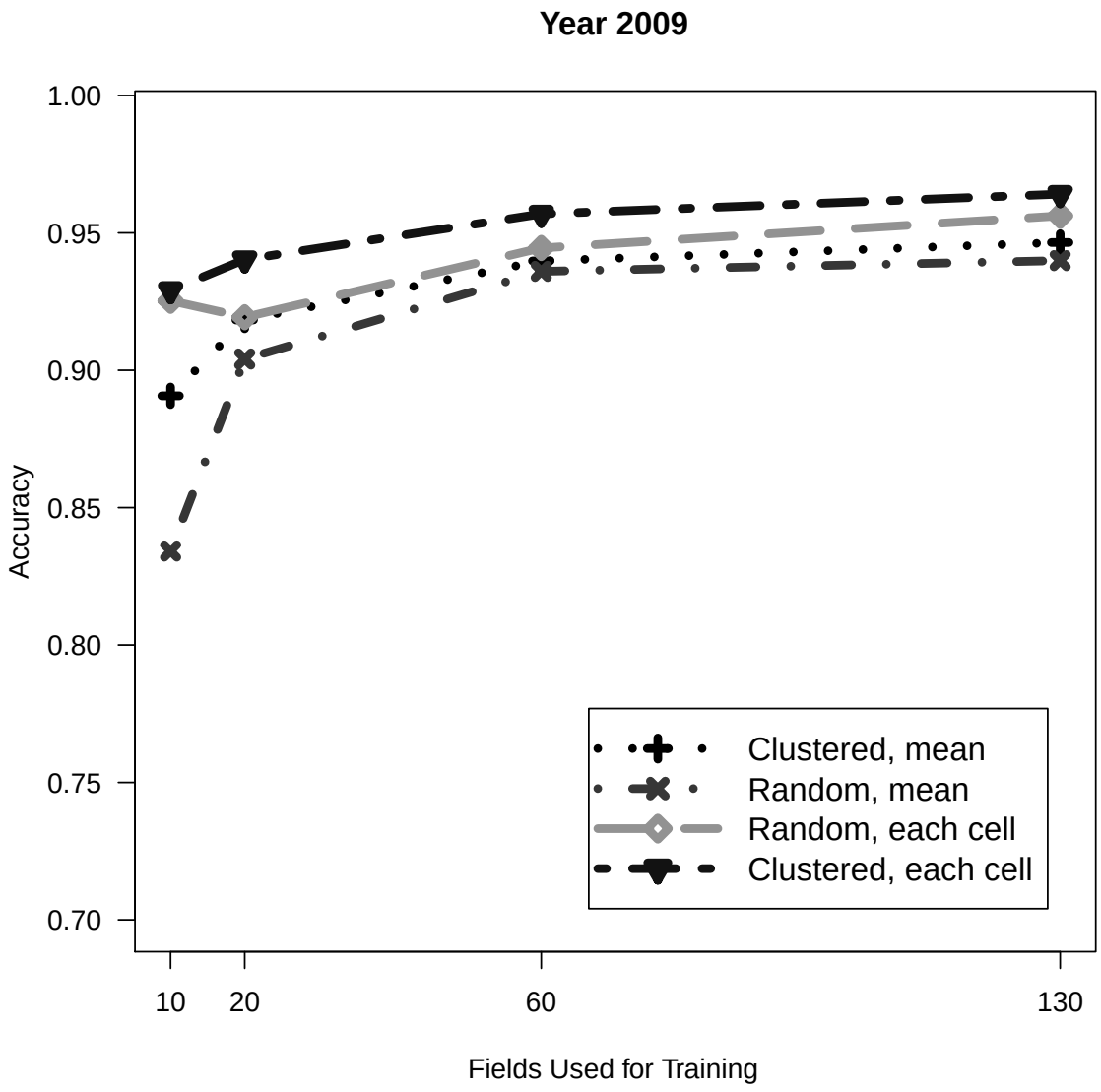

Figure 4.6. This year has the best overall performance. The accuracies are higher across the board than any other years. It has the latest starting image, but has 5 images that cover the growing season evenly. The difference between the top two methods is very close, and even the worst performing catch up quickly as the number of fields is increased.

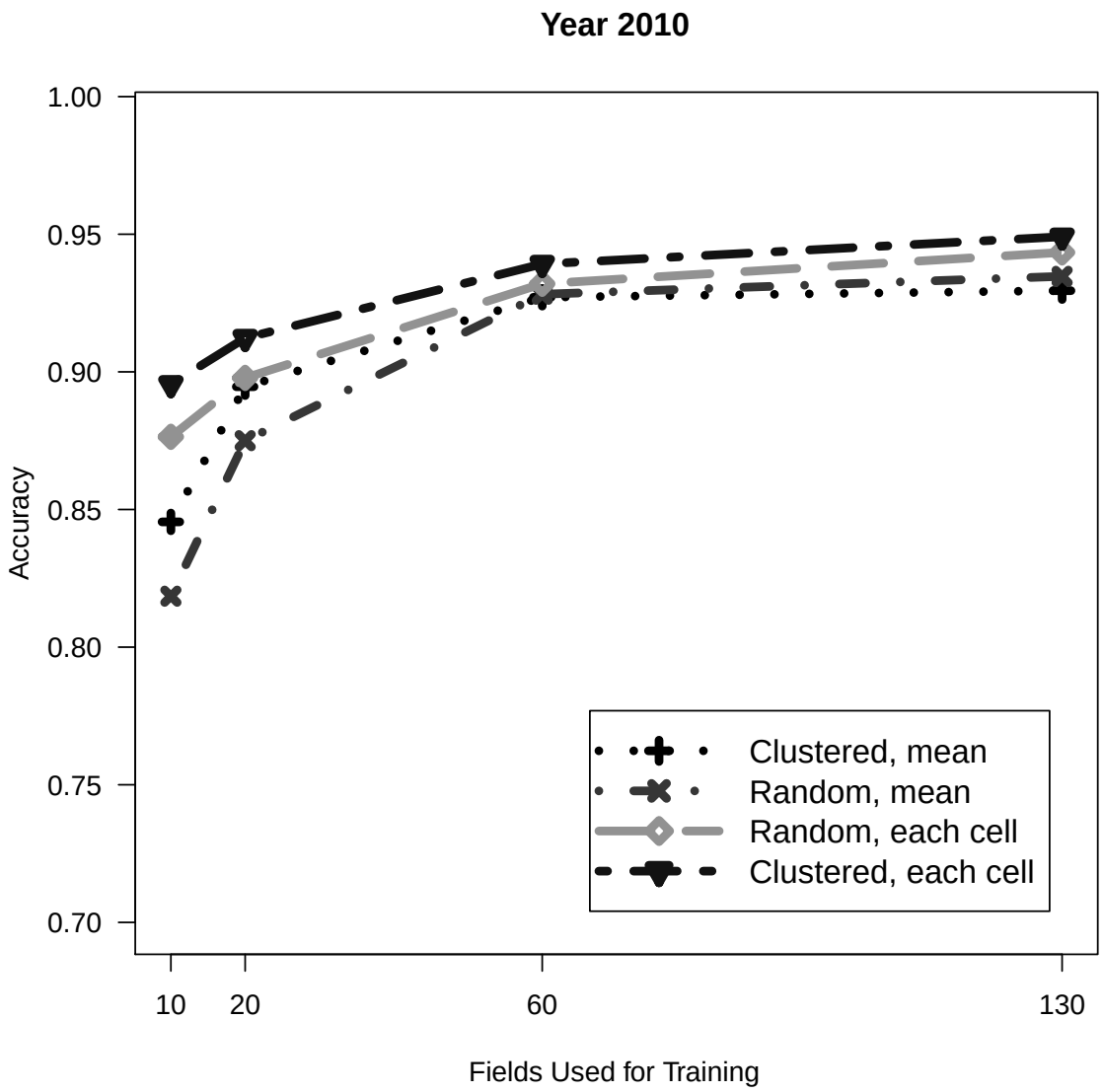

Figure 4.7. In terms of performance, this year is right in the middle. It has the most images, at 6, and covers the whole growing season well. All of the methods have slightly decreased accuracy compared to the best performing years. Even so, the best performing, clustered each cell, outperforms random each cell by almost 2 percentage points. The fact that all the accuracies are depressed points to an anomaly in this year's data.

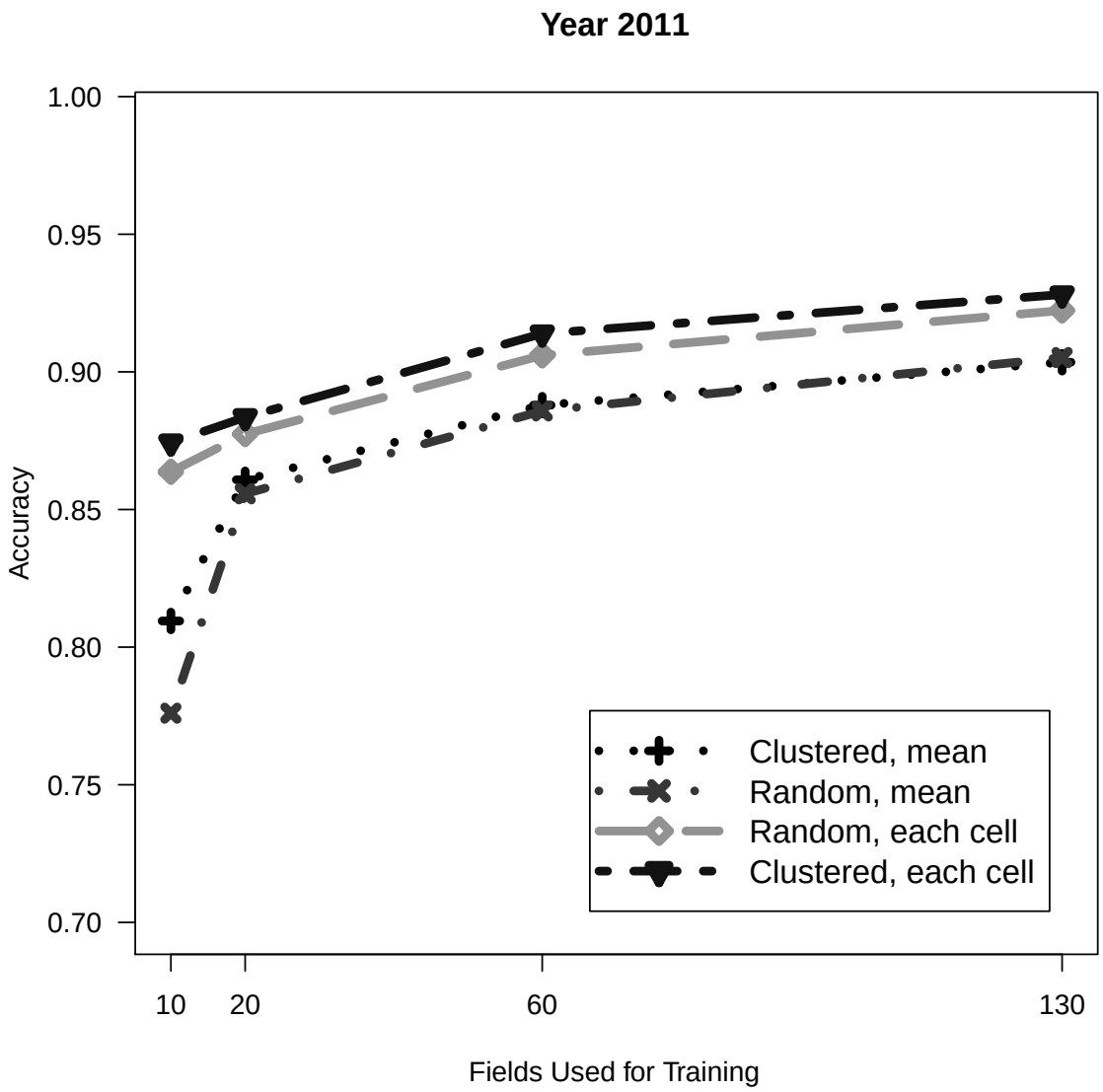

Figure 4.8. This year is the second worst performing. As discussed in the analysis of 2008 (see Figure 4.5), this year has only 4 images. It has an earlier starting image, by 23 days, and a much later ending image, but the gap between image 3 and image 4 is over 60 days. Furthermore, the ending image is so late that most beet fields are harvested. This is why this year is slightly better than the worst, but not much.

5. CONCLUSIONS

We have shown three things. First, classifying individual members of a set and then aggregating the labels in order to label the set performs better overall than using the mean value of the set. Secondly, clustering the sets and choosing an equal number of sets from each cluster gives better accuracy than selecting a training set by random sampling when dealing with a very small training set. The final point is that clustering produces a larger effect when using the mean values than it does when classifying individual set members. The upshot of this is that in a situation where only mean values are available, or using individual set members is impractical, our method of clustering can give an improvement in accuracy over choosing random members for training.

REFERENCES

[1] Sugato Basu, Mikhail Bilenko, and Raymond J Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68. ACM, 2004.

[2] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. KNIME: The Konstanz Information Miner. In *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*. Springer, 2007.

[3] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. 2004.

[4] Begüm Demir, Luca Minello, and Lorenzo Bruzzone. A cost-sensitive active learning technique for the definition of effective training sets for supervised classifiers. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 1781–1784. IEEE, 2012.

[5] Rob Emanuele. Geotrellis. *http://geotrellis.io/*, Accessed February, 2014.

[6] GRASS Development Team. *Geographic Resources Analysis Support System (GRASS GIS) Software*. Open Source Geospatial Foundation, 2012.

[7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[8] David J. Hand and Keming Yu. Idiot's bayes: Not so stupid after all? *International Statistical Review / Revue Internationale de Statistique*, 69(3):pp. 385–398, 2001.

[9] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[10] A Huete, K Didan, T Miura, E.P Rodriguez, X Gao, and L.G Ferreira. Overview of the radiometric and biophysical performance of the {MODIS} vegetation indices. *Remote Sensing*

of Environment, 83(1–2):195 – 213, 2002. The Moderate Resolution Imaging Spectroradiometer (MODIS): a new generation of Land Surface Monitoring.

[11] Richard R. Irish. Landsat 7 automatic cloud cover assessment. volume 4049, pages 348–355, 2000.

[12] Richard R Irish, John L Barker, Samuel N Goward, and Terry Arvidson. Characterization of the landsat-7 etm+ automated cloud-cover assessment (acca) algorithm. *Photogrammetric engineering and remote sensing*, 72(10):1179, 2006.

[13] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 03 1951.

[14] Julie B. Odenweller and Karen I. Johnson. Crop identification using landsat temporal-spectral profiles. *Remote Sensing of Environment*, 14(1-3):39 – 54, 1984.

[15] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[16] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. ISBN 3-900051-07-0.

[17] Brian D Wardlow and Stephen L Egbert. State-level crop mapping in the us central great plains agroecosystem using modis 250-meter ndvi data. In *Pecora 16 Symposium*, pages 25–27, 2005.