Interactively Training Pixel Classifiers

Justus H. Piater and **Edward M. Riseman** and **Paul E. Utgoff**

Computer Science Department
University of Massachusetts
Amherst, MA 01003
lastname@cs.umass.edu

Abstract

Manual generation of training examples for supervised learning is an expensive process. One way to reduce this cost is to produce training instances that are highly informative. To this end, it would be beneficial to produce training instances interactively. Rather than provide a supervised learning algorithm with one complete set of training examples before learning commences, it would be better to produce each new training instance based on knowledge of which instances the learner would otherwise misclassify. Whenever the learner receives one or more new training examples, it should update its classifier incrementally and, in real time, provide the teacher with feedback about its current performance. The feasibility of such an approach is demonstrated on a realistic image pixel classification task. Here, the number of training instances involved in building a classifier was reduced by several orders of magnitude, at no perceivable loss of classification accuracy.

Introduction

As advances are made in technology for machine learning, one can expect to see this technology incorporated in tools for constructing decision making components of larger systems that non-specialists build. In particular, pixel classifiers are an important component of many vision applications, e.g. texture-based segmentation (du Buf, Kardan, & Spann 1990; Blume & Ballard 1997), image understanding (Campbell *et al.* 1997; Jolly & Gupta 1996), object recognition (Hafner & Munkelt 1997), obstacle detection (Langer & Jochem 1996), and geoscience (Carr 1996; Carpenter *et al.* 1997).

Despite these abundant applications, the construction of high-performance pixel classifiers usually involves substantial cost in terms of human effort. A traditional procedure for classifier construction is illustrated in Figure 1: A number of training instances (i.e. completely or partially hand-labeled images) are selected and fed to a classifier construction system. The resulting classifier is then evaluated, typically by comparing its output with ground truth data and assessing its accuracy. If the performance is not satisfactory, some parameters of the system are changed, such as the feature set or the training set, or the classifier construction algorithm, and the entire procedure is repeated.

It is well known that the appropriateness of the training set has a great influence on the performance of a classifier. For this reason, significant effort is traditionally put into the construction of the training set. This work is concerned with efficient selection of informative training instances. In the case of image pixel classification, substantial cost is incurred by the requirement to provide by hand the correct labels for the training pixels. Therefore, one would like to be able to provide a small number of well chosen training instances relatively quickly, at no loss of classification accuracy, or even improved performance (Salzberg *et al.* 1995).

There are other benefits to keeping the training set small. For example, a typical decision tree classifier will make every attempt to place training instances of different classes in separate leaf nodes, as long as they are discernible based on their feature vectors. However, in most practical applications the distributions of different classes overlap in feature space, which leads to overly specialized and very complicated decision trees with poor generalization properties. This is typically addressed by elaborate pruning algorithms which try to detect overspecialization and then simplify the decision tree. Such pruning reduces the classification accuracy on the training set to some degree, but in practice the accuracy on independent test data often increases. In essence, classification accuracy on the training set is traded for improved generalization. Other types of classifiers address this problem differently, e.g. by drawing maximum-likelihood boundaries between classes in feature space. To generate optimal classifiers, such algorithms require a sufficiently large number of training instances whose distributions in feature space meet the statistical assumptions made by the algorithm. In many practical applications this requirement cannot be met.

Consequently, it would be beneficial to select a *small* number of *informative* training instances that are known to be typical representatives of their class, rather than a large number from an unknown distribution. In the case of deci-

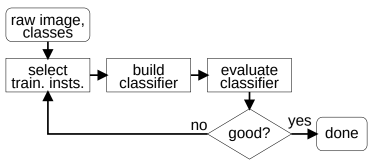


Figure 1: Traditional classifier construction.

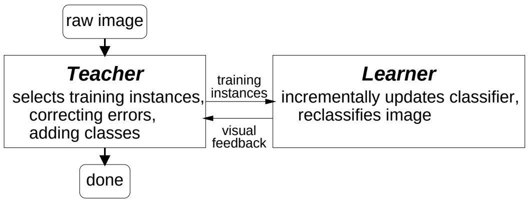


Figure 2: The On-Line Classifier framework: Interactive, incremental classifier construction.

sion tree classifiers, such a procedure should ideally eliminate the need for pruning altogether.

This raises the question of what constitutes a well chosen training instance. If one could know where the classifier currently makes mistakes, one could generate an informative instance by providing a correct label for a currently misclassified pixel. We propose an interactive system for efficient construction (in terms of human involvement) of pixel classifiers. In our system, the off-line iterative procedure (Figure 1) is replaced by an interactive incremental Teacher-Learner paradigm (Figure 2), which we call the On-Line Classifier. The Teacher is a human domain expert who operates a graphical user interface. He can select images for training and, for any image, select and label small clusters of pixels. The Learner is a computer program that operates through a well-defined communication interface with the Teacher's interface. The Learner can receive images and training instances, and can quickly produce a classifier and labels for the pixels of the current training image, according to the most recent classifier.

A fundamental aspect of this model is that it is incremental. The Teacher does not need to provide a large number of instances that may or may not be informative. Instead, each time the user provides a new instance, the Learner rapidly revises its classifier as necessary, and then communicates the class labels for all pixels of the image. This lets the user see the misclassified pixels with almost no delay. He can immediately respond by providing correct labels for one or more of them, which are passed as new training examples to the classifier. In this sense, we call a newly supplied training instance *informative* if and only if it is misclassified by the current classifier.

Incremental Decision Trees

This work is primarily concerned with effective selection of training instances. Another important issue in classifier construction is the selection of a feature set. It is known that increasing the size of a feature set can adversely affect classifier performance (Devijver & Kittler 1982). Selection of an optimal feature subset from a given universe of features has been shown to be infeasible in practice (Ferri *et al.* 1994). Classifiers that utilize all available features (such as neural networks, nearest-neighbor clusterers, linear machines) are particularly sensitive to redundant and noisy features. This motivates the use of a univariate *decision tree* classifier which consults only a single feature at each decision node. Only *useful* features are incorporated into the tree, and features of little discriminative power are disregarded entirely. "Useful" here refers to the ability of the classifier to classify the training set correctly. If overfitting is effectively avoided by proper selection of training instances, then the resulting decision tree may not require all available features. One is still left with the problem of selecting representative training instances that will cause the tree induction algorithm to select those features that will result in good generalization. Thus, we have not solved the feature selection problem, but by employing an interactive decision tree paradigm we can address it in terms of training instance selection.

With the On-Line classifier, the user presents training instances sequentially to the classifier construction system, and expects the classifier to incorporate each new training example very quickly. Thus, the system requires a classifier that can be rebuilt or incrementally upgraded very quickly, without unlearning previously learned instances. This rules out many classifiers, e.g. neural networks which converge relatively slowly and require a large number of training example presentations. Decision trees, on the other hand, are known for their computational efficiency.

An early incremental decision tree algorithm was proposed by Crawford (1989) based on CART (Breiman *et al.* 1984). When a new training instance would cause a new test to be picked at a decision node, the entire subtree rooted at this node is discarded and rebuilt based on the corresponding subset of the training examples. Lovell and Bradley (1996) constructed another partially incremental decision tree algorithm, the "Multiscale Classifier". It works by adjusting decision thresholds and, if necessary, splitting leaves by introducing new decision nodes. Because all the data seen are not stored in the tree, these adjustments may cause previously processed instances to be classified incorrectly. Therefore, these instances must be presented to the decision tree again, which in turn may cause alterations of the tree. The method refines the tree incrementally, and the result is dependent on the order of the training instances.

The Incremental Tree Inducer ITI (Utgoff 1994; Utgoff, Berkman, & Clouse 1997) solves this problem by storing all data relevant for restructuring a decision tree within the nodes (Schlimmer & Fisher 1986). It can accept and incorporate training instances serially without needing to rebuild the tree repeatedly. Another desirable characteristic is that it produces the same tree for the same accumulated set of training instances, regardless of the order in which they are received. It can also operate in conventional batch mode, where the full set of training instances is made available at once. The classification accuracy is statistically indistinguishable (Utgoff, Berkman, & Clouse 1997) from that of C4.5 (Quinlan 1993), which is widely considered one of the leading decision tree algorithms.

One drawback of univariate decision trees like ITI is that decision boundaries best described by functions of multiple features must be approximated by multiple univariate decisions. Nevertheless, for the On-Line Classifier, ITI's computational efficiency (in terms of tree revision and instance classification) and relatively good generalization properties make it an excellent system. It achieves a very quick feedback loop, consisting of receiving a new training instance, updating the classifier, and reclassifying the image.

To maximize the utility to the user, pixels near the location of the latest training pixel are (re)classified first and displayed by our graphic user interface. The user can select new training pixels at any time, allowing very rapid training even on large images without delay.

Qualitative Discussion

This section walks through an example session shown in Figure 3. The goal is to learn to classify pixels as one of SKY, ROOF, BRICK, or FOLIAGE. Pixels that belong to another region type, e.g. PAVEMENT, are not of interest. None of these pixels will be labelled by the teacher, and will therefore never serve as a training instance. Six features are used, which are the red/green/blue measurements of a pixel, and the variances of each in a 3×3 window centered around that pixel. Each mouse click of the teacher produces a 3×3 square of training instances that is used to update the learner's decision tree.

Figure 3b shows the classification result after labeling just one square of each of two classes. The sky is already almost perfectly separated from the rest of the image. In Figures 3c and d, one square of each of the remaining two classes is added. While the addition of BRICK again results in good generalization, things become more complicated when a sample of the FOLIAGE class is added. This occurs in this image because FOLIAGE and ROOF are mainly characterized by large variances within the RGB intensities rather than the colors themselves, and thus hard to separate. The image in Figure 3h contains several contradictory training instances that belong to different classes (FOLIAGE and ROOF) but are indistinguishable using the given feature set. Therefore, perfect classification is not achievable, given these features.

Quantitative Results

We now compare performance of our On-Line Classifier with a previously published classification result by Wang et al. (Wang *et al.* 1997). We chose this example because it uses state-of-the-art techniques, the task is realistic, and their data include ground truth.

Wang et al. considered a monochromatic aerial image (1,936,789 pixels) of a rural area in Ft. Hood, Texas (Figure 5a). The goal was to build a pixel classifier to recognize the four terrain classes BARE GROUND (road, riverbed), FOLIAGE (trees, shrubs), GRASS, and SHADOW. Their most effective feature set consisted of 12 *co-occurence features* (angular second moment, contrast, and entropy at four angular orientations each (Haralick, Shanmugam, & Dinstein 1973)), four *three-dimensional features* (Wang *et al.* 1997), and the gray value. The co-occurence features employed have previously been claimed to be highly effective for classification (Conners & Harlow 1980; du Buf, Kardan, & Spann 1990; Ohanian & Dubes 1992; Weszka, Dyer, & Rosenfeld 1976). The 3D features are generated during stereo processing of a calibrated image pair (Schultz 1995) and were recently shown to be highly discriminative in this task (Wang *et al.* 1997). The Foley-Sammon transform (FST, Foley & Sammon 1975) was employed as a classifier. FST is a linear discriminant method

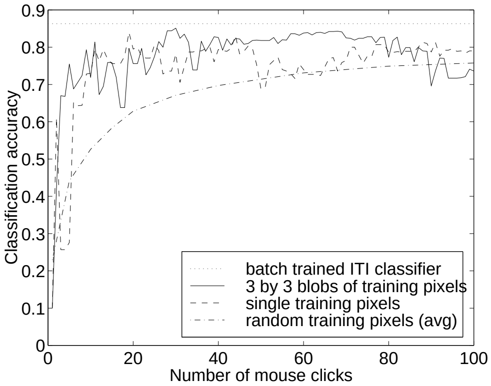


Figure 4: Plot of classification accuracy versus number of mouse clicks during interactive training of a classifier. Each mouse click generated either a 3×3 square of training pixels, or a single training pixel. The ITI batch classifier trained on Wang et al.'s 16916 training pixels and an average of 100 runs with randomly selected training pixels are also shown. For the latter curve, each training pixel was picked from each class with equal probability as a human might do, even is a class is rare.

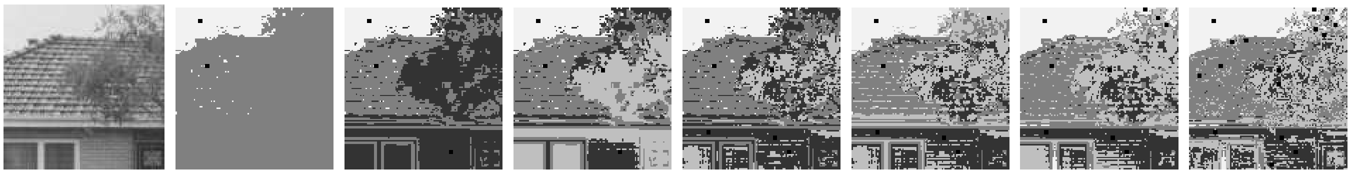
that is considered effective (Liu, Cheng, & Yang 1993; Weszka, Dyer, & Rosenfeld 1976).

As a training set, Wang et al. used four homogeneous square regions of different sizes: 99×99 (FOLIAGE), 75×75 (GRASS), 37×37 (BARE GROUND), and 11×11 (SHADOW). This was one of their best training sets found after extensive experimentation. The 16916 training pixels constitute less than 1% of the entire image (1,936,789 pixels). Ground truth was generated by hand. The achieved classification accuracy is 83.4% (Wang *et al.* 1997).

To provide a baseline of the performance of ITI with respect to FST on this task, we ran ITI in conventional batch mode on the same input data as described above, using the full training set of 16916 pixels. ITI achieved a classification accuracy of 86.3% (86.4% using ITI's pruning mechanism), outperforming FST.

We then trained a classifier interactively on this data, using the On-Line Classifier. The intermediate decision trees resulting from each mouse click were saved and subsequently used to generate a performance curve. Figure 4 shows that excellent classification accuracy was achieved after very few mouse clicks. On the other hand, the accuracy achieved by Wang et al.'s training set of 16916 pixels was not quite reproduced within the first 100 mouse clicks. This shows that continued training should yield further improvement in the long run. However, one must bear in mind that the evaluation is done on a single image. Continuing to select training instances from this image will lead to a very specialized classifier with poor generalization properties for other images. This is likely to be the case with the preselected training set.

This point is best illustrated by a brief analysis of some of the resulting decision trees. Table 1 shows that batch training on the large preselected training set produced a large tree which employed nearly all available features, even after



a b c d e f g h

Figure 3: An example training session: (a) grayscale version of the original color image (112 by 115 pixels); (b)–(d): results after adding one set of training instances for each class; (e)–(h): snapshots during some refining. 3×3 squares of training pixels appear as tiny black squares. Legend: SKY, ROOF, FOLIAGE, BRICK.

	interactive		*batch*	
			unpruned	*pruned*
# Mouse clicks:	19	30		
# Train. insts.:	19	270	16916	16916
% correct:	84.0	85.1	86.3	86.4
# tree nodes:	9	25	143	71
# features used:	3	7	15	15

Table 1: Summary of classification results using ITI. The total number of features available is 17.

pruning. On the other hand, the interactively trained classifiers were both very small and used only small subsets of the available features, at very little loss in classification accuracy (see also Figure 5)! Undoubtedly the complex trees accounted for a large number of exceptions that cannot be expected to generalize to other (similar) images, while the simple classifiers achieved good results because their very few training pixels were *selected in an informed manner*.

For comparison with uninformed selection of training pixels, Figure 4 includes a learning curve of a classifier trained by randomly selected training pixels, regardless of whether a newly chosen training pixel is misclassified by the current classifier. This curve rises much more slowly than the interactively built classifiers. Clearly, informed selection of training examples can facilitate the rapid construction of simple decision tree classifiers.

It is also interesting to note that there is a component of human skill in selecting useful training examples. Figure 6 depicts a learning curve created by selecting training pixels at random from among currently misclassified pixels only. This implies that each new training pixel alters the classifier and is therefore *informative* according to our definition. In fact, this learning curve rises somewhat faster than if pixels are selected purely at random. However, it still does not even come close to a learning curve trained by a human teacher. At some point – after about 40 training pixels – the curves cross. (Even though there is much variability in the random learning curves, this phenomenon is statistically significant.) A possible explanation for this is that after this point, most "typical" representatives of a class are already classified correctly, and forcing the algorithm to select a currently misclassified pixel causes overspecialization by including atypical "exceptions" into the tree.

Conclusion

We have demonstrated a new interactive methodology for training of pixel classifiers. It is a very effective tool for selecting few but informative training instances, resulting in great reduction of human labor and dramatically simplified classifiers. Real-time feedback provided through a simple user interface allows rapid training, on the order of a few minutes in our realistic example. The efficiency of the feedback loop is not limited by the size of the training image.

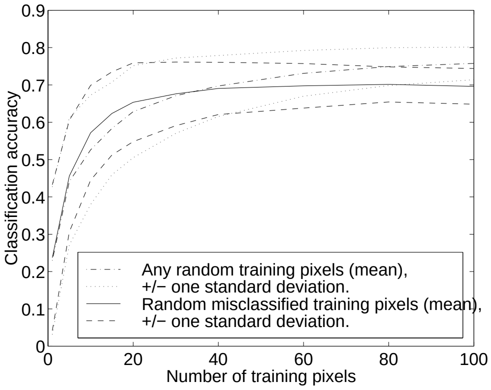


Figure 6: Plot of classification accuracy versus number of randomly selected training pixels. Each curve represents an average of 100 individual runs. The dash-dotted curve is identical to the one shown in Figure 4. The 95% confidence interval is around 0.02.

To build interactive learning systems that update their parameters in real time, incremental learning algorithms are beneficial. The ITI classifier was chosen because of its capability to incorporate training instances incrementally, and because of the implicit feature selection property of decision trees. While it works well in our applications, more experiments with this and other classification algorithms will be performed on more complex tasks. Fast incremental learning algorithms are an open area of research with many potential applications for interactive learning systems.

Acknowledgements

The sample implementation makes use of the official ITI distribution which is accessible over the internet at `http://www-ml.cs.umass.edu/iti/`. We thank X. Wang for providing the feature files and ground truth data for the Ft. Hood imagery. This research has been supported in part by the Advanced Research Projects Agency (via Army Research Labs) under contracts DAAL02-91-K-0047 and DACA76-97-K-0005, and by the National Science Foundation under grant IRI-9222766.

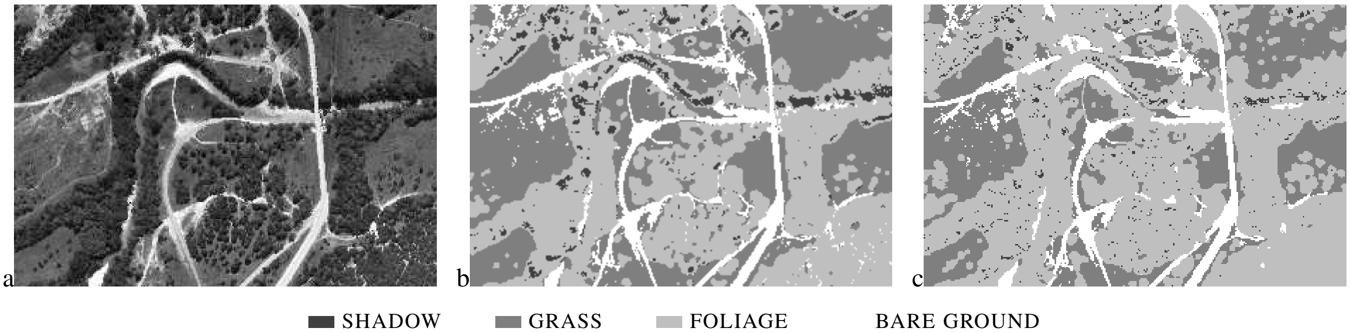


Figure 5: Subset of the Ft. Hood scene (300 × 200 pixels): (a) aerial image; (b) classification results using ITI trained in batch mode on the same training set as in Wang et al.; (c) classification results using ITI with interactive incremental training. The two classification results are very similar; differences mainly occur in regions that are ambiguous even to a human.

References

Blume, M., and Ballard, D. R. 1997. Image annotation based on learning vector quantization and localized Haar wavelet transform features. *Proc. SPIE* 3077:181–190.

Breiman, L.; Friedman, J. H.; Olshen, R. A.; and Stone, C. J. 1984. *Classification and regression trees*. Pacific Grove, CA: Wadsworth&Brooks.

Campbell, N. W.; Mackeown, W. P. J.; Thomas, B. T.; and Troscianko, T. 1997. Interpreting image databases by region classification. *Pattern Recognition* 30(4):555–563.

Carpenter, G. A.; Gjaja, M. N.; Gopal, S.; and Woodcock, C. E. 1997. ART neural networks for remote sensing: vegetation classification from Landsat TM and terrain data. *IEEE Trans. Geoscience and Remote Sensing* 35(2):308–325.

Carr, J. R. 1996. Spectral and textural classification of single and multiple band digital images. *Computers & Geosciences* 22(8):849–865.

Conners, R., and Harlow, C. 1980. A theoretical comparison of texture algorithms. *IEEE Trans. Pattern Anal. Machine Intell.* 2(3):204–222.

Crawford, S. L. 1989. Extensions to the CART algorithm. *Int. J. Man-Machine Studies* 31:197–217.

Devijver, P. A., and Kittler, J. 1982. *Pattern recognition: a statistical approach*. Englewood Cliffs: Prentice-Hall.

du Buf, J.; Kardan, M.; and Spann, M. 1990. Texture feature performance for image segmentation. *Pattern Recognition* 23(3/4):291–309.

Ferri, J. J.; Pudil, P.; Hatef, M.; and Kittler, J. 1994. Comparative study of techniques for large-scale feature selection. In Gelsema, E. S., and Kanal, L. N., eds., *Pattern Recognition in Practice IV*, 403–413. Elsevier Science B.V.

Foley, J. D., and Sammon, Jr., J. 1975. An optimal set of discriminant vectors. *IEEE Trans. on Computers* 24(3):281–289.

Hafner, W., and Munkelt, O. 1997. Using color for detecting persons in image sequences. *Pattern Recognition and Image Analysis* 7(1):47–52.

Haralick, R.; Shanmugam, K.; and Dinstein, I. 1973. Textural features for image classification. *IEEE Trans. Systems, Man, and Cybernetics* 3(6):610–621.

Jolly, M.-P. D., and Gupta, A. 1996. Color and texture fusion: application to aerial image segmentation and GIS updating. In *IEEE Workshop on Applications of Computer Vision*, 2–7.

Langer, D., and Jochem, T. 1996. Fusing radar and vision for detecting, classifying and avoiding roadway obstacles. In *Proc. IEEE Intelligent Vehicles Symposium*, 333–338.

Liu, K.; Cheng, Y.; and Yang, J. 1993. Algebraic feature extraction for image recognition based on an optimal discriminant criterion. *Pattern Recognition* 26(6):903–911.

Lovell, B. C., and Bradley, A. P. 1996. The multiscale classifier. *IEEE Trans. Pattern Anal. Machine Intell.* 18(2):124–137.

Ohanian, P., and Dubes, R. 1992. Performance evaluation for four classes of textural features. *Pattern Recognition* 25(8):819–833.

Quinlan, J. R. 1993. *Programs for machine learning*. Morgan Kaufmann.

Salzberg, S.; Delcher, A.; Heath, D.; and Kasif, S. 1995. Best-case results for nearest-neighbor learning. *IEEE Trans. Pattern Anal. Machine Intell.* 17(6):599–608.

Schlimmer, J. C., and Fisher, D. 1986. A case study of incremental concept induction. In *Proc. Fifth Nat. Conf. on Artificial Intelligence*, 496–501. Philadelphia, PA: Morgan Kaufmann.

Schultz, H. 1995. Terrain reconstruction from widely separated images. *Proc. SPIE* 2486:113–123.

Utgoff, P. E.; Berkman, N. C.; and Clouse, J. A. 1997. Decision tree induction based on efficient tree restructuring. *Machine Learning* 29(1):5–44.

Utgoff, P. E. 1994. An improved algorithm for incremental induction of decision trees. In *Machine Learning: Proc. 11th Int. Conf.*, 318–325. Morgan Kaufmann.

Wang, X.; Stolle, F.; Schultz, H.; Riseman, E. M.; and Hanson, A. R. 1997. Using three-dimensional features to improve terrain clasification. In *Proc. Computer Vision and Pattern Recognition*, 915–920.

Weszka, J.; Dyer, C.; and Rosenfeld, A. 1976. A comparative study of texture measures for terrain classification. *IEEE Trans. Systems, Man, and Cybernetics* 6(4):269–285.