Appears in the Proceedings of the Fourteenth International Conference on Machine Learning (ICML-97), Nashville, TN.

Feature Engineering and Classifier Selection: A Case Study in Venusian Volcano Detection

Lars Asker*
Jet Propulsion Laboratory
M/S 525-3660
Pasadena, CA 91109-8099
asker@aig.jpl.nasa.gov

Richard Maclin
Department of Computer Science
University of Minnesota
Duluth, Minnesota 55812-2496
rmaclin@d.umn.edu

Abstract

As machine learning has graduated from toy problems to "real world" applications, users are finding that "real world" problems require them to perform aspects of problem solving that are not currently addressed by much of the machine learning literature. Specifically, users are finding that the tasks of selecting a set of features to define a problem and obtaining a set of examples of the problem are often more important for a successful machine learning application than the selection or development of a specific classification method. In this paper we present a case study of machine learning applied to a difficult "real world" problem: detecting volcanos in SAR (synthetic aperture radar) images of Venus from the Magellan dataset. Our work demonstrates that the processes of feature selection and sample collection are critical to the production of a good classifier. We further show that the use of domain dependent knowledge can often serve to enhance the resulting classifier. Finally, we demonstrate that an ensemble approach to building a classifier, where multiple component classifiers are used in combination, makes the issue of selecting a "best" classification method moot since the ensemble outperforms any of the individual component classifiers.

1 INTRODUCTION

The successful utilization of machine learning techniques for "real world" applications often requires a developer to exercise a completely different set of skills than are required for applying machine learning to "toy world" domains. Toy domains often supply a pre-existing problem formulation (pre-classified training examples) and representation (pre-determined set of features), leaving the user only the job of selecting or developing an appropriate learning algorithm; our work suggests that a successful "real-world" application of machine learning is an iterative process which focuses on problem formulation and representation and leaves the selection of algorithm as a later detail. We also show that rather than selecting a single classifier method, an ensemble of different classifiers can be used to produce a classifier that often outperforms any single classifier. Furthermore we suggest that a successful "real world" application often requires the utilization of a collection of domain dependent "tricks" as well as a strong understanding of the domain.

We will exemplify these ideas and show how they have been applied in the development of a system that learns to recognize small volcanos in SAR (synthetic aperture radar) images of Venus collected by the Magellan spacecraft (Saunders 1992). This problem is of interest because it is important scientifically and because the huge volume and the high dimensionality of the data (images) makes it very difficult. The huge volume of data prevents manual analysis of the dataset and motivates an automated approach.

2 "REAL WORLD" DIFFICULTIES

The application of machine learning to "real world" problems highlights aspects of the development process that are not currently addressed in much of the machine learning literature, a few exceptions are (Brodley & Smyth in press) and (Langley & Simon 1995). Good feature engineering often turns out to be more important for producing good classification accuracy in a learning system than the selection of a particular classification algorithm. The feature engineering step involves several difficulties.

---

* Current address: Department of Computer and Systems Sciences, Stockholm University

One difficulty is how to formulate the problem as a learning problem and what constitutes an example (or counterexample). In the present application this turned out to be especially challenging. Expert geologists who were consulted to label a portion of the available data showed considerable disagreement in their individual labelings. This was handled by using a consensus approach based on combining several experts labelings, producing a "ground truth" labeling (Burl *et al.* 1996; 1994).

Another difficulty with "real world" applications is the very high number of features that are possible for representing the data (or the high dimensionality of the problem). This is especially true when working with image data. Most such problems are simply not feasible to solve with existing ML methods. Instead, the number of features has to be reduced before any classification algorithm can be applied. We have two approaches to this problem, each one aimed at reducing the number of features. The first is to use principal component analysis (PCA) (Fukunaga 1990; Jolliffe 1986) to come up with a small number of linear combinations of the original features that are still sufficient to describe the examples. The other is to let different classifiers use different subsets of features and then to produce an ensemble of classifiers that combines the output from the different component classifiers.

A third difficulty is the presence of outliers or noise in the training data. There is little prior work on the detection of natural objects in a noisy environment. Most image detection algorithms work only on man-made objects which have very sharp boundaries. The noisy features of natural objects makes these methods difficult to apply. In the current application the presence of noise is handled by finding and replacing suspected noisy pixel values before the principal component analysis step.

3 THE PROBLEM: VOLCANO DETECTION

The study of volcanism on Venus is of particular interest to planetary geologists since it is the most widespread and important geologic phenomenon of that planet (Saunders 1992). Understanding the clustering characteristics and global distribution of the volcanos is fundamental to understanding the regional and geological evolution of the planet (Crumpler *et al.* 1997; Guest 1992). Even a partial catalog of the planet including the size, location, and other relevant information about each included volcano would clearly be helpful for more advanced study. Such a catalog could potentially provide the data necessary to answer basic questions about the geophysics of Venus such as the relationship between volcanos and local tectonic structure, the pattern of heat flow within the planet, and the mechanics of volcanic eruption.

The Magellan spacecraft, which was launched by NASA/JPL in May of 1989, has provided the scientific community with a set of more than 30,000 synthetic aperture radar (SAR) images covering 98% of the surface of Venus. This data set is larger than what has been produced from all previous space probes combined and has motivated an automated (or at least semi-automated) approach to analyzing the dataset. Figure 1 contains a sample of part of an image from the Magellan SAR dataset. The SAR images are 1024 by 1024 pixel images with a resolution of 75m per pixel. It has been estimated that there are on the order of one million small (less than 20 km in diameter) volcanos visible in the Magellan images (Aubele & Slyuta 1990). A catalog of large Venus volcanos has been completed (Crumpler *et al.* 1997; Stofan 1992) but, by optimistic estimates, the time for a geologist to manually generate a comprehensive catalog of small volcanos on Venus is on the order of ten to twenty man-years due to the size of the dataset.

To add to the problem of detecting volcanos, even the experts do not completely agree on the location of the volcanos in the images that have been examined. This is understandable, because while the images are fairly high resolution, there are a number of cases that are difficult to judge and there is currently no other means to verify any experts' decisions. This is a problem in that classification methods generally depend on having a set of pre-classified training instances to produce a classifier for unseen instances. To address this problem a group of experts was gathered together under the auspices of the Jet Propulsion Laboratory (JPL) to attempt to produce a "ground-truth" labeling of some of the images. This was an exhausting process involving each expert labeling the set of images examined on their own and then meetings amongst the experts to produce a consensus labeling of the images. As a result a first baseline for measuring the performance of any method is the expert performance with respect to the consensus. For each expert we can measure how many of the "actual" (according to the consensus) volcanos that expert correctly labeled. We can also measure the number of non-volcanos (again according to the consensus) that the expert produced. The tradeoff between these two values (the number of hits versus the number of false positives) will be the focus of our results.

3.1 JARTOOL

Even with computer classification, the amount of data in the dataset is extremely large. The JARtool system (Burl *et al.* 1994; 1996) which is based on machine

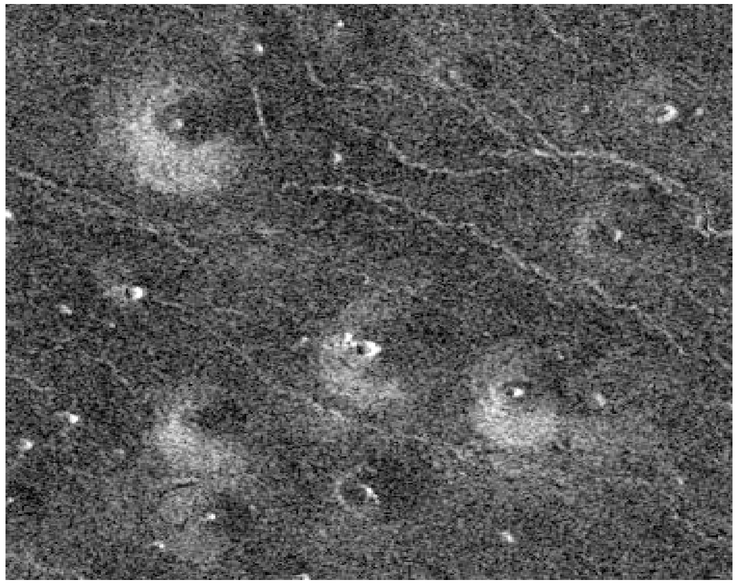

Figure 1: Sample of part of an image from the Magellan SAR image set. The image shows a small region that contains a number of volcanos.

learning and pattern recognition techniques and was developed by JPL, is a system developed for this classification process. JARtool is trained by first filtering the data in a pre-pass method called the Focus Of Attention (FOA). The FOA is a simple method that uses a matched filter for selecting particular sized areas (usually squares) of the image that are more likely to contain a volcano. This model has two positive effects: (1) it greatly reduces the number of data points to consider; and (2) it causes each volcano to be roughly "centered" in the sub-image. The first effect is very important since even when a relatively small sub-image (say 15 pixels by 15 pixels) is used to recognize volcanos, the resulting sub-image still has a large number of features (225 pixel values in this case). The major disadvantage of using the FOA model is that by pre-selecting a small number of sub-images from the original image the resulting set of sub-images may not include all of the volcanos labeled by the experts.

Once the FOA model has been applied the problem of volcano detection is one of determining which of the sub-regions of the image returned by the FOA actually contain volcanos.

The original JARtool method controlled for the high-dimensional space using the principal-component analysis method (discussed in Section 4.1) to extract a reduced set of features.

After the dimensionality reduction step, the resulting features were then used to train a quadratic (or Gaussian) classification method to distinguish between actual volcanos and non-volcanos. In the production phase, the Gaussian classifier gives the probability that any region that matches the filter is an actual volcano.

Results for the JARtool method are shown in Figures 4 and 5. Note that due to the use of the FOA model the resulting classifier has an upper limit in its accuracy that is less than 100%, since some of the actual volcanos are left out. While the performance of this method is good, there may be some room for improvement since experts outperform this method. We have therefore extended the original method in a number of ways which contribute to increased classification accuracy of the total system. Each of these improvements will be described in the following sections.

4 MOTIVATION AND BACKGROUND

We make use of a number of techniques to format our problem as a machine learning problem and to produce a classifier for the resulting problem. In this section we will in turn describe: Principal component analysis, clustering of volcanos, handling of noisy pixels, and using ensembles of classifiers.

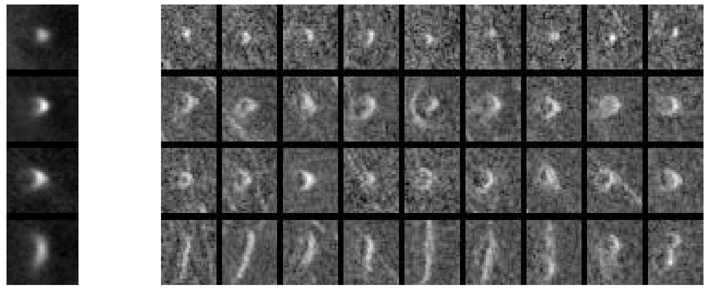

Figure 2: Example volcanos from 4 different clusters (right) and their respective cluster centers (left). Each row represents a sample of volcanos that have been clustered together using K-means.

4.1 AUTOMATIC FEATURE EXTRACTION

To produce a volcano detector our algorithm must be able to label a set of small images as being either volcanos or not volcanos. Since these sub-images consist of a large number of pixels, the resulting input space has high dimensionality, and the set of possible features becomes immense. We have therefore restricted our search to the family of features defined by linear combinations of the image pixel values.

One "solution" to the problem of projecting a high-dimensional space onto a more tractable low-dimension space is to use principal components analysis (PCA) (Fukunaga 1990). The method of principal components has been used extensively in statistics, signal processing (Karhunen-Loeve transform), and pattern recognition (Turk & Pentland 1991).

In PCA one forms the covariance matrix C for the example feature vectors, and finds the eigenvalues and eigenvectors of C. The m eigenvectors having the largest eigenvalues are then used as the new features. This strategy is equivalent to projecting the n-dimensional pixel space onto a q-dimensional subspace (feature space).

PCA can be shown to be optimal in a least-squares sense for representing the example feature vectors with minimal loss of information (Fukunaga 1990; Jolliffe 1986). Unfortunately, it does not always provide the best linear combination of features for discriminating between the different classes since the distribution of the "other" class is not taken into consideration.

Other approaches, such as linear discriminant analysis (LDA), seek to find *discriminative* features that separate the classes. In the context of finding volcanos, however, the "other" class is amazingly complex consisting of all patterns that are not volcanos. Direct application of LDA in pixel space leads to very poor results. One drawback with LDA is that it produces a maximum of $c-1$ features (where c is the number of classes). In our case, there are only two classes so a reduction to a single linear combination of the pixel values is clearly inadequate.

4.2 FINDING SUBTYPES OF VOLCANOS

One weakness with the scheme proposed in JARtool is the fact that it is based on the assumption that all volcanos look enough alike to be selected by a single filter in the FOA step and to be classified by a single classifier. In practice, there exists a variety of subtypes of volcanos, each with its own visual characteristics. Figure 2 shows some different types of volcanos. The volcanos in each row in the right side of the figure are taken from different clusters while the left side of the figure displays the corresponding cluster center for each row. Instead of training one single classifier to distinguish between typical volcanos and non-volcanos, it makes sense to train a collection of different classifiers (each with its own particular set of features) and hence hopefully create a collection of classifiers that each have their own area of expertise. In order to do so we first used k-means clustering to partition the volcanos in the training data into a number of clusters in a way that minimizes the sum of the squared distances to the cluster centers. Each such cluster was then fed to the principal component analysis step described in Subsection 4.1 to produce a set of features that best describes that particular cluster.

4.3 HANDLING OF NOISE AND OUTLIERS

The detection of natural objects in images presents a number of problems. Often these images contain significant noise and do not contain sharp features. Many current approaches assume that the location of the object has been marked and simply focus on the problem of discriminating between a set of possible objects. One approach that has been tried is the use of Hough transforms to locate features closely resembling circles in SAR data (Cross 1988;

Skingley & Rye 1987). For volcano location, Wiles and Forshaw developed a matched-filter approach that has been applied to Magellan data (Wiles & Forshaw 1993), but this method has limited performance.

To handle outliers we have developed a more robust mechanism for the feature learning component that allows for the presence of non-typical feature values in positive training examples. This mechanism includes finding the mean and standard deviation for each pixel position for a set of volcano images. Then for each pixel position in each image we determine how much the value deviates from the mean value for that position. If it is further away than a constant times the standard deviation of all the values for that position, then the value is replaced by another value which is closer to the mean (the mean added or subtracted by the constant times the standard deviation). Parameter sensitivity tests has shown that the use of this method alone will increase the classification accuracy by 1-2 %.

4.4 ENSEMBLES OF CLASSIFIERS

An ensemble is a classifier created by combining the predictions of multiple component classifiers. A number of researchers have demonstrated that ensembles are generally more accurate than any of their component classifiers (Breiman 1996; Clemen 1989; Quinlan 1996; Wolpert 1992; Zhang, Mesirov, & Waltz 1992). Figure 3 shows a basic framework for combining classifiers. Using an ensemble, the class of an example is predicted by first classifying the example with each of the component classifiers and then combining the resulting predictions into a single classification. To create an ensemble a user generally must focus on two aspects: (1) which classifiers to use as components of the ensemble; and (2) how to combine the resulting predictions into a single prediction.

Much research on selecting appropriate classifiers to combine has focused on selecting classifiers that are accurate in the predictions, but differ in where they are accurate. Methods for approaching this problem include using different classification methods, training on subsets of the data set, training on different sets of input features, using different subsets of the training set for training the classifiers, and even using genetic search to try to find classifiers that disagree in their predictions (Breiman 1996; Drucker *et al.* 1994; Hansen & Salamon 1990; Hashem, Schmeiser, & Yih 1994; Krogh & Vedelsby 1995; Maclin & Shavlik 1995; Opitz & Shavlik 1996). The method of choosing different classification methods is interesting, since most classification methods introduce particular biases into the resulting classification. Also appealing is the method of varying the set of input features, since the resulting component classification problems in fact differ significantly when the features are not completely redundant.

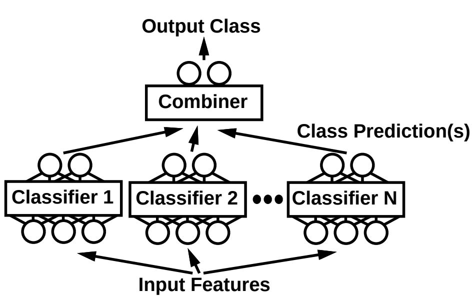

Figure 3: Basic framework for combining multiple classifiers.

The second aspect of creating an ensemble is the choice of the function for combining the predictions of the component classifiers (Kearns & Seung 1995). Examples of combination functions include voting schemes (Hansen & Salamon 1990), simple averages (Lincoln & Skrzypek 1989), weighted average schemes (Perrone & Cooper 1994; Rogova 1994), and schemes for *training* combiners (Rost & Sander 1993; Wolpert 1992; Zhang, Mesirov, & Waltz 1992). Clemen demonstrated that in the absence of knowledge concerning a specific problem, almost any reasonable method, including the simple ones such as voting or using a weighted average will result in an effective ensemble (Clemen 1989). It is possible, though, to make use of knowledge about a specific domain to produce a more accurate combination method (Rost & Sander 1993; Zhang, Mesirov, & Waltz 1992). Our method applies each of the best classifiers in sequence starting with the best. If any classifier gave a probability above a certain threshold then that classifier was applied, otherwise the decision was left to the rest of the classifiers. If none of the best classifiers could reach a decision, the default method of using the simple average of all the classifiers was applied. This will be described in more detail in Section 5.2.

5 EXPERIMENTS

Below we present the experiments we performed to demonstrate the validity of our approach.

5.1 EXPERIMENTAL METHODOLOGY

To develop our classifier we started out by examining a set of four images that have been labeled and examined in previous work (Burl *et al.* 1994; 1996). We used these images as a means for evaluating which combinations of features and classification methods to use. These four images contain 163

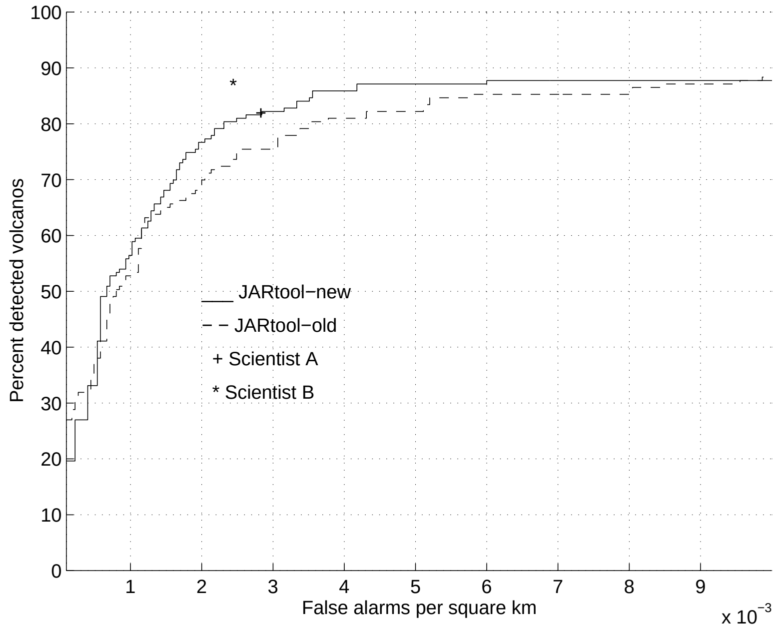

Figure 4: Results from our experiments, the original JARtool method and two experts on the original set of four images. Results are graphed by the number of misclassified non-volcanos allowed per km^2. As the number of allowable misclassified non-volcanos increases the total percentage of actual volcanos increases, though this value has an upper limit less than 100% because the FOA model only recognizes a certain percentage of the actual volcanos.

volcanos, 144 of which are included in the set of regions located by the FOA model. The FOA model also produces 481 sub-images which match the filter but are not volcanos. To produce a classifier for each of these images we trained an ensemble of classifiers using the other three images as the training set (i.e., four-fold cross-validation using the volcanos/non-volcanos of each image as a fold). Once we had chosen the different combinations of features and classifiers we intended to use in our classifier we then tested our method on a set of 38 labeled images separate from the original set of four images. These images contain 453 volcanos, 383 of which are recognized by the FOA model. The FOA model also produces 9920 sub-images that are not volcanos. To produce classifiers for these images we divided the images up into six sets and performed six-fold cross-validation on each of these sets.

For all of our results we show a curve with the percentage of detected true positives relative to the number of detected false positives per km^2. This is done by successively lowering the threshold for predicting a volcano and determining how many true volcanos are included versus how many "false positive" volcanos are included (i.e., as the threshold lowers more of the actual volcanos are included, but more "false positives" may appear).

5.2 DESIGN DETAILS

We began our study by focusing on selecting an initial feature representation. Preliminary work with classifying the data based on the raw pixel information produced very poor results due to the high dimensionality of the problem and the (relatively) small number of examples compared to the number of features. Therefore, we started by assuming we would apply PCA to the volcano and non-volcano sub-images (separately) returned by the FOA. We then performed experiments on the preliminary set of four images varying the size and scaling (averaging over several pixels to produce a small image of a large area) of the sub-images returned by the FOA model and the number of principal components and found that the following combinations using principal components only from the volcanos seemed to work well for a simple Gaussian classifier: principal components: 6, 8, and 10; scaling: 2, 3, and 4. To the set of principal component features we added two further features based on knowledge of the domain and PCA: (1) a line filter value that notes the presence of lines in the image – these lines can easily distract the FOA model; and (2) the reconstruction error which is an indication of how much information is lost when a particular image is projected onto the first n principal components.

After selecting the initial set of features we then explored the approaches of replacing noisy pixels and doing clustering of the images before applying PCA, again varying the number of principal components used to characterize the images. For the noisy pixel replacement we tried different values of the threshold used to determine when to replace a pixel. When using clustering, we performed clustering on the volcano images and then selected the top n principal components across all of the clusters as features for the sub-image. For the noisy replacement method we found that the following thresholds worked well: 0.25, 0.45, and 0.65 standard deviations. For the clustering we found that 1 and 4 clusters worked well.

The values for all the parameters were determined after extensive parameter sensitivity tests, some of which can be found in (Burl *et al.* 1996). However, it should be noted that this system, like most real-world classification systems, consists of multiple components, each with their own parameters and that an overall system optimization is not practically possible. Instead many parameters were set to reasonable values based on univariate parameter sensitivity testing.

Once we selected the different sets of features to use we approached the selection of the classification method. We have experimented with a variety of algorithms including quadratic (or Gaussian) classifiers, decision trees, linear discriminant analysis, nearest neighbors using Euclidean and spatially weighted distance measures (Turmon 1996), tangent distance (Simard, Cun, & Denker 1993), kernel density estimation, Gaussian mixture models, and feed-forward neural networks (Cherkauer 1996).

Interestingly enough, all of these methods (with the exception of linear discriminant analysis) gave similar performance on an initial data set, indicating that the selection of features is much more important than the selection of classifier(s). In the experiments reported in this paper a combination of the quadratic classifier and feed-forward neural networks have been used. Both classifiers provide posterior probability estimates, which can be thresholded to vary the trade-off between detection and false alarm rate, a feature we considered important to be able to combine the classifiers in a uniform way.

In our initial work we noted that the Gaussian classifiers were extremely accurate for cases where the output value was extremely high or low. We therefore determined to develop an ensemble method that would first look at the output value and if the value fell above a threshold 0.999 we would mark the example as a volcano and if it fell below a threshold 0.01 we would mark it as a non-volcano.

The Gaussian methods seemed to be very good at determining that some small subsets of the images either were or were not volcanos, but for intermediate cases we decided to add a second classification method – neural networks. To produce our neural network methods we trained a group of 48 networks where we varied the input features to each method:

- The best 6, 8, and 10 principal components with and without reconstruction errors plus line filter values. (6 networks)
- The best 6, 8, and 10 principal components with and without reconstruction errors plus line filter values for volcanos where noisy pixel replacement was performed at levels of 0.25, 0.45, and 0.65. (18 networks)
- The best 10, 12, 14, and 16 principal components from clustered volcanos with and without reconstruction errors plus line filter values with noisy pixel replacement at thresholds of 0.25, 0.45, and 0.65 (24 networks)

From this group of 48 neural networks we created one single classification method based on the simple average of all 48 networks. This resulting method was then used to classify any examples not meeting the threshold criteria described above.

5.3 EXPERIMENTAL RESULTS

The above described classification method produced results which are shown in Figure 4. The resulting classifier outperforms all of the component classifiers, the original JARtool method (Burl *et al.* 1994; 1996) and even outperforms one of the experts. Of course, these results are for a dataset where we have performed significant exploration to select input features, etc. so it is not surprising that we perform well.

To test our method we trained the same set of classification methods using the same thresholds on a set of 38 unseen images. The results from these images are shown in Figure 5.

Again, our approach outperforms the original JARtool method along most of the ROC curve, but this time our method falls slightly below expert performance. Also in this case, the combined classifier outperforms each one of the individual component classifiers (their performance have been left out of Figures 4 and 5 for clarity).

For a description of an automatic way to determine the thresholds see (Asker & Maclin 1997).

6 CONCLUSIONS

We have presented a case study for the application of machine learning to solve "real world" problems.

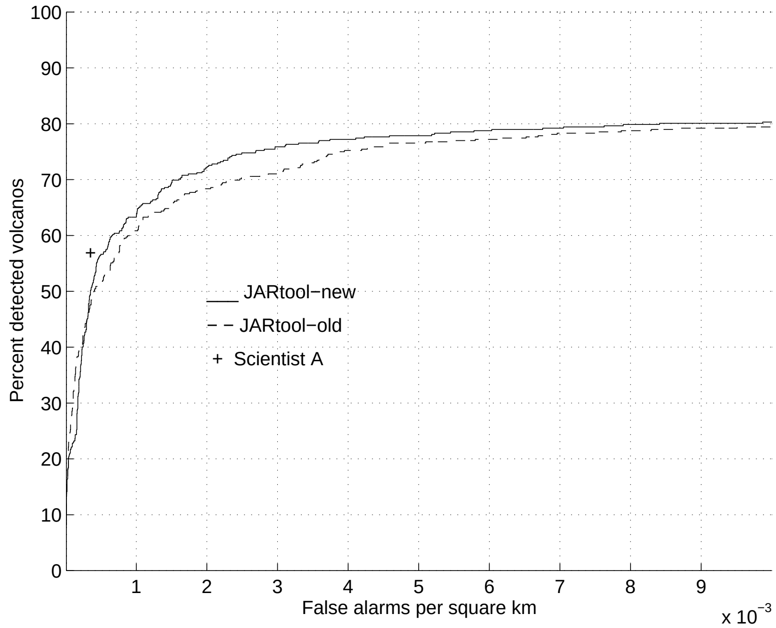

Figure 5: Results from our experiments, the original JARtool method and one expert on the new set of 38 images.

Our research demonstrates that the issues of framing a problem (feature selection) and gathering data are critical for the development of a classification method, often more critical than the selection of the actual learning algorithm. In our work we showed the usefulness of a number of approaches to feature selection. In particular, we explored the use of principal component analysis to map a high dimensional feature space to a low dimensional feature space. We also demonstrated the use of techniques such as replacement of noisy information and clustering of examples before applying principal component analysis as useful methods for good feature extraction. Our work also indicates that the use of domain-specific knowledge can serve to greatly improve the quality of the resulting features used to represent a problem.

As our method of classification we chose to employ an ensemble approach. The advantage of this approach is that we do not have to settle on a particular classification method or a particular set of features, but can combine multiple methods to produce a classifier that outperforms any individual classification method. To produce our ensemble we again made use of domain knowledge (in this case, a set of preliminary data) in forming our ensemble. This preliminary data allowed us to select a function for combining the component classifiers that took advantage of the abilities of each of the component classifiers.

The result of our approach is a method for Volcano detection from SAR images of Venus returned by the Magellan probe that outperforms all previous methods and produces near-expert performance for this difficult problem.

Acknowledgments

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration (NASA). The work has been partially supported by a grant from the Swedish Research Council for Engineering Sciences (L. Asker) and by a NASA/ASEE Summer Faculty Fellowship (R. Maclin). We would like to thank P. Stolorz, M. Burl, P. Smyth, and U. Fayyad for their assistance with the Volcano dataset.

References

Asker, L., and Maclin, R. 1997. Ensembles as a sequence of classifiers. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*.

Aubele, J. C., and Slyuta, E. N. 1990. Small domes on venus: Characteristics and origins. *Earth, Moon and Planets* 50/51:493–532.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

Brodley, C. E., and Smyth, P. in press. Applying classification algorithms in practice. *Statistics and Computing*.

Burl, M.; Fayyad, U.; Perona, P.; Smyth, P.; and Burl, M. 1994. Automating the hunt for volcanos on venus. In *Proceedings of the 1994 Computer Vision and Pattern Recognition Conference*, 302–309.

Burl, M.; Fayyad, U.; Perona, P.; and Smyth, P. 1996. Trainable cataloging for digital image libraries with application to volcano detection. Technical Report Computation and Neural Systems, Caltech.

Cherkauer, K. 1996. Personal communication.

Clemen, R. 1989. Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting* 5:559–583.

Cross, A. 1988. Detection of circular geological features using the hough transform. *International Journal of Remote Sensing* 9:1519–1528.

Crumpler, L. S.; Aubele, J. C.; Senske, D. A.; Keddie, S. T.; Magee, K. P.; and Head, J. W. 1997. Volcanoes and centers of volcanism on venus. In Bougher, S. W.; Hunten, D. M.; and Phillips, R. J., eds., *Venus II*. University of Arizona Press.

Drucker, H.; Cortes, C.; Jackel, L.; LeCun, Y.; and Vapnik, V. 1994. Boosting and other machine learning algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning*, 53–61. New Brunswick, NJ: Morgan Kaufmann.

Fukunaga, K. 1990. *Introduction to statistical pattern recognition*. New York: Academic Press. second edition.

Guest, J. E. 1992. Small volcanic edifices and volcanism in the plains of venus. *Journal of Geophysical Research* 97:15949–15966.

Hansen, L., and Salamon, P. 1990. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12:993–1001.

Hashem, S.; Schmeiser, B.; and Yih, Y. 1994. Optimal linear combinations of neural networks: An overview. In *Proceedings of the 1994 IEEE International Conference on Neural Networks*.

Jolliffe, I. T. 1986. *Principal component analysis*. New York: Springer-Verlag.

Kearns, M., and Seung, H. 1995. Learning from a population of hypotheses. *Machine Learning* 18:255–276.

Krogh, A., and Vedelsby, J. 1995. Neural network ensembles, cross validation, and active learning. In Tesauro, G.; Touretzky, D.; and Leen, T., eds., *Advances in Neural Information Processing Systems*, volume 7. Cambridge, MA: MIT Press.

Langley, P., and Simon, H. A. 1995. Applications of machine learning and rule induction. *Communications of the ACM* 38:55–64.

Lincoln, W., and Skrzypek, J. 1989. Synergy of clustering multiple back propagation networks. In Touretzky, D., ed., *Advances in Neural Information Processing Systems*, volume 2, 650–659. San Mateo, CA: Morgan Kaufmann.

Maclin, R., and Shavlik, J. 1995. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*.

Opitz, D., and Shavlik, J. 1996. Generating accurate and diverse members of a neural-network ensemble. In Touretsky, D.; Mozer, M.; and Hasselmo, M., eds., *Advances in Neural Information Processing Systems*, volume 8. Cambridge, MA: MIT Press.

Perrone, M., and Cooper, L. 1994. When networks disagree: Ensemble method for neural networks. In Mammone, R. J., ed., *Artificial Neural Networks for Speech and Vision*. London: Chapman and Hall.

Quinlan, J. R. 1996. Bagging, boosting, and c4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*.

Rogova, G. 1994. Combining the results of several neural-network classifiers. *Neural Networks* 7:777–781.

Rost, B., and Sander, C. 1993. Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology* 232:584–599.

Saunders, R. S. 1992. Magellan mission summary. *Journal of Geophysical Research* 97:13067–13090.

Simard, P.; Cun, Y. L.; and Denker, J. 1993. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems 5*, 50–258.

Skingley, J., and Rye, A. 1987. The hough transform applied to sar images for thin line detection. *Pattern Recognition Letters* 6:61–67.

Stofan, S. 1992. Distribution and characteristics of coronae and related features on venus. *Journal of Geophysical Research* 97:13347–13378.

Turk, M., and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3:71–86.

Turmon, M. 1996. Personal communication.

Wiles, C., and Forshaw, M. 1993. Recognition of volcanoes using correlation methods. *Image and Vision Computing* 11:188–196.

Wolpert, D. 1992. Stacked generalization. *Neural Networks* 5:241–259.

Zhang, X.; Mesirov, J.; and Waltz, D. 1992. Hybrid system for protein secondary structure prediction. *Journal of Molecular Biology* 225:1049–1063.