Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Machine Learning despite Unknown Classes

Christopher B. Smith
Southwest Research Institute
San Antonio, Tx, USA
cbsmith@ieee.org

***Abstract*—This paper revisits supervised machine learning for multiclass problems with the assumption that all classes cannot be represented in a training set. This is common in many applications in which there are numerous classes or in which some classes are exceedingly rare. In this paper we propose the use of a decision function to serve in place of the decision boundaries which are used in many machine learning techniques. We demonstrate this technique using Fisher's iris data and an application to language recognition.**

***Index Terms*—Machine learning, multiclass machine learning**

I. Introduction

The supervised machine learning literature contains many techniques for the classification of multiple classes. Techniques such as the support vector machine (SVM), probabilistic neural networks, generalized discriminant analysis are each at their core binary classifiers. A few notable exceptions include, multiclass extensions for decision tree learning [4][5], and specialized versions of boosting such as AdaBoost.M2 and AdaBoost.MH, [7] [8] and a version of SVM[6]. With a few exceptions most multiclass approaches are binary techniques which are combined to form multiple decisions, for example one-versus-all and one-versus-one voting strategies commonly used with SVM[1][2].

This dominance of binary techniques is largely due to the theoretical niceties of binary classification. It is easier to measure sensitivity and specificity, and more generally error, when the choice is between two classes. For a finite number of known classes, it is also reasonable to assume that each of the decision boundaries needed for classification can be constructed using binary decision boundaries. From linear system theory this is very analogous to applying superposition to the problem of constructing decision boundaries. Each boundary can be constructed from a sum of many other boundaries. These are long-held principles which are correct given a few reasonable assumptions about the classification problem.

One of these assumptions is that all classes must be known ahead of time, in other words the classes are available as part of the training data. While many problems can be posed with only a limited number of known classes, many cannot. For example, full scale face recognition, or the ability to recognize and uniquely classify all faces, is not practical as all 6.7 billion human faces cannot be included in a single database.

This problem of unknown classes also occurs when some forms of the object to be recognized are exceedingly rare, or are costly to collect data about. Such a scenario occurs often in security applications. For example, machine learning techniques are often used in detection problems, such as network intrusion detection. In network intrusion detection, data is largely available for detecting conditions such as denial of service attacks. Unfortunately data simply does not exist for unknown network attack techniques.

The remainder of this paper presents simple a technique to approach learning unknown classes. We term this technique a "decision function" as opposed to the decision boundary or separating hyperplane used in binary classification. The next section formalizes the decision function. The following two sections present examples using Fisher's Iris data and a language recognition problem. Finally the paper is ended with some conclusions and comments on the case of unknown classes.

II. The Decision Function

Supervised machine learning as found in [1][2], etc. can be described as follows. Given empirical data of two forms, measurements $\mathbf{x}$ and a class label y for each set of measurements. The simple case is that of two classes:

$$\{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathbb{R}^n, y_i \in \pm 1\}$$

We also define a distance metric using either linear or kernel-based projection techniques.

$$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$$

A simple binary classifier can be constructed by

$$y = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i k(x, x') + b\right) \tag{1}$$

By optimizing the choice of α_i and b a decision surface can be devised.

This approach to machine learning is then extended to the multiclass case by computing multiple binary classifications. For cases where all classes cannot be defined in the training set, this form of learning becomes awkward. A binary test can be used to identify unknown vs. known classes, but this technique requires a data rich definition of what is "unknown" to allow training. For many applications this kind of scheme has been acceptable. [10] [11]

Here we propose a different approach, rather than defining a discrete set of decision boundaries, we propose a "continuum" of decision. A decision is made via a second function used in place of the sgn in (1).

The idea here is simple, we follow the same procedure of projecting down from a high dimensional space onto a single dimension, but rather than compute a simple decision based on

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

a set of known classes, we apply a function which can result in a class assignment other than those present in the training data. We leave "extra" classes latent in the decision function.

This function could have any number of forms. An example of such a decision function could be as simple as:

$$y = \text{round}\left(\sum_{i=1}^{m} \alpha_i k(x, x') + b\right)$$

or more generally:

$$y = \text{f}\left(\sum_{i=1}^{m} \alpha_i k(x, x') + b\right)$$

Unfortunately we have taken a problem with parameters α_i and b and have added an additional parameter f which will complicate computing all three. The challenge becomes determining some relationship between classes which allows for the function f to be learn-able.

III. Fisher's Iris Data

In [**?**], Fisher analyzes data on three species of Iris flowers to determine a technique for distinguishing them. He chose: Iris setosa, Iris versicolor, and Iris virginica. An alternative formulation could be of to recognize **any** subtype of the genus Iris, whether a known species or hybrid between species. This task is much more difficult. There are as many as 300 known species of Iris. Additional species can be created by hybridizing two of these known species. Even these new species could be recognized and classified by a knowledgeable human observer, but for current machine learning techniques this debatably straightforward task is difficult. This requires an exhaustive training set including all species.

In [**?**], classification is performed based on the four measured features: sepal length, sepal width, petal length, and petal width. Fisher settles on the following projection weights for making a decision

$$\mathbf{w}_f = \begin{bmatrix} -3.308998 \\ -2.759132 \\ 8.866048 \\ 9.392551 \end{bmatrix}$$

where a decision could be based on the following

$$y = \sum_{i=1}^{m} w_f(i) x(i)$$

The scatter plot of the features and resulting projection y are shown in figure 1.

To actually make the decision, one could do one of two things (1) assume discrete set of points to break up this continuous space, for example 30 and 6 (the two points which are exactly between means),

$$f = \begin{cases} \text{y} > 30 & \mathbf{x} \in \text{Virginica} \\ \text{y} < 6 & \mathbf{x} \in \text{Setosa} \\ \text{otherwise} & \mathbf{x} \in \text{Versicolor} \end{cases} \quad (2)$$

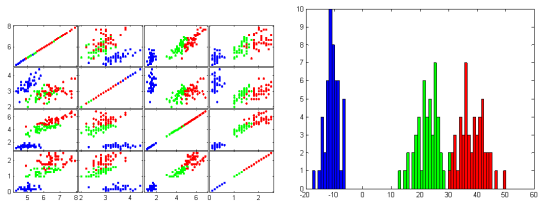

Fig. 1. Distribution of Fisher's Iris data (left) features, (right) projected with weights $\mathbf{w}_f$.

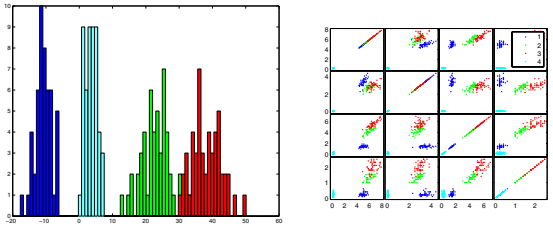

Fig. 2. New iris data (left) projected using the weights from $\mathbf{w}_f$, and (right) scatter plot of new features.

or (2) one could build a function such as

$$f = \text{round}\,(y)$$

which implicitly computes the class identification. In this case the class identification is in the form of an integer.

In (2) above, this extreme discretization of f would give up the ability to extend this same transformation to a new set of classes. For example, new simulated measurements were created of a new species of iris, with much smaller features. The histogram and scatterplot of the projection of this new species is shown in figure 2.

By discretizing f less severely than in (2) for example using,

$$f = \text{round}\left(\tfrac{1}{10}y + 0.6\right)$$

The new species of iris can be easily identified despite the lack of samples in the training set. In this case the new species would lie in class "0", while setosa, versicolor, and virginica would lie in -2, 1, and 3 respectively.

IV. Language Recognition

In the WALS, World Atlas of Language Structures online [9], over 2000 languages are indexed with features measured about each. In the WALS, many languages like English and Mandarin are extensively studied and highly defined, but others such as the native American or South Pacific languages are spoken by so few people, little is known about them. Given such a database, a language problem can be formulated as "given a set of training data build a classifier which can recognize *any* language", even those not represented in the training set.

To begin, we begin by selecting a subset of 23 languages which have extensive features measured. These included English, Mandarin, Japanese, French, German, and Spanish. We

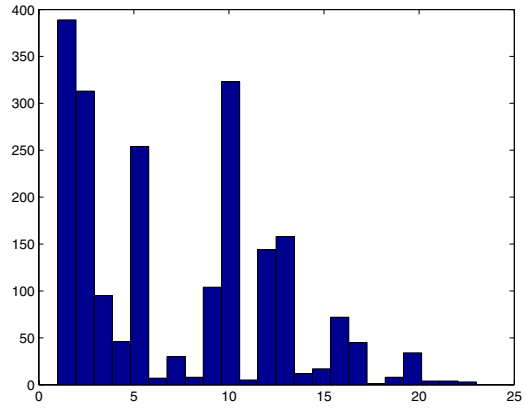

Fig. 3. Projection of 2076 languages into 23 classes, resulting in 2053 errors.

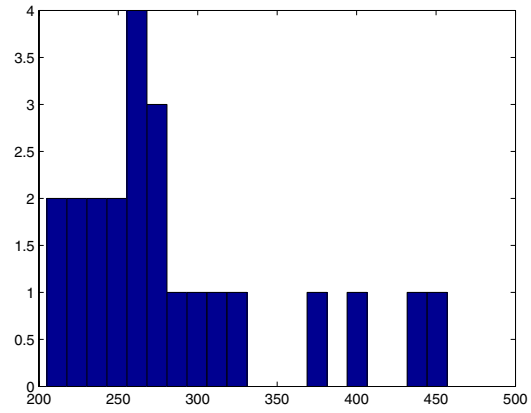

Fig. 4. Distribution of 23 classes across the integers, shown with a histogram bin size of 12.6.

use the perceptron algorithm to train a linear classifier to correctly classify each of these languages. We then applied this classifier to a set of 2076 languages. The result is shown in Figure 3. This naturally results in 2053 errors.

Instead of using 22 discrete decision boundaries, we use a decision function which maps the 23 classes to the set of integers. The distribution of the 23 classes is shown in figure 4. Using this simple decision function which requires no further training data, the 2076 classes, as shown in figure 5 are now mapped to 279 separate classes. This is an improvement from 23 correct classifications to 279 correct classifications. This is without any extra information added or assumed in training.

V. Conclusion

This paper addresses a common though not well-studied problem, supervised machine learning when all classes are not represented in the training set. This is an inconvenient problem which does not decompose well into a set of decision

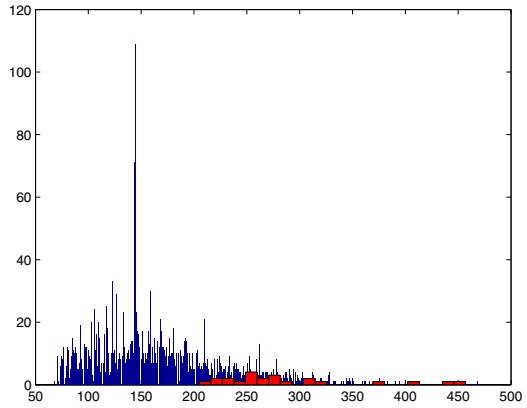

Fig. 5. Projection of 2076 languages onto the integers with the original 23 classes, resulting in 279 correct classifications.

boundaries. Rather it requires projecting the training set onto a function to predict relationships between classes. This complication changes the nature of the classification problem, but does not invalidate the existing techniques. Instead it adds a new dimension to the problem, resulting in a function rather than a set of decision boundaries.

Acknowledgment

This work was supported by a Southwest Research Institute. The author would like to thanks the many who have contributed comments.

References

[1] B. Scholkopf, A. Smola, *Learning with kernels*, MIT Press, 2002.
[2] R. Duda, P. Hart, D. Stork, *Pattern Classification*, Wiley-Interscience, 2001.
[3] R. Fisher, The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, (7): 179-188, 1936.
[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, 1984.
[5] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
[6] K. Crammer, Y. Singer, On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines, *Journal of Machine Learning Research*, 265-292, 2001.
[7] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119139, August 1997.
[8] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):140, 1999.
[9] M. Haspelmath, M. Dryer, D. Gil and B. Comrie, *WALS Online*, Munich: Max Planck Digital Library, 2008.
[10] V. Chandola,A. Banerjee, and V. Kumar, Anomaly Detection: A Survey, *ACM Computing Surveys*, 2009.
[11] T. Ahmed, B. Oreshkin and M. Coates, Machine Learning Approaches to Network Anomaly Detection, *USENIX*, 2007.