Learning by Combining Native Features with Similarity Functions

EXTENDED ABSTRACT

Mugizi Rwebangira(rweba@cs.cmu.edu)

October 31, 2008

1 Introduction

The notion of exploiting data dependent hypothesis spaces is an exciting new direction in machine learning with strong theoretical foundations[Shawe-Taylor et al.(1998)Shawe-Taylor, Bartlett, Williamson, and Anthony]. A very practical motivation for these techniques is that they allow us to exploit unlabeled data in new ways [Balcan and Blum(2006)]. In this work we investigate a particular technique for combining "native" features with features derived from a similarity function. We also describe a novel technique for using unlabeled data to define a similarity function.

2 Learning with Generic Similarity Functions

Our work is a direct development of the work of [Balcan and Blum(2006)]. In their work they show that it is possible to use a similarity function which is not necessarily a legal kernel to explicitly map data into a new space such that if the data was separable by a similarity function with a certain margin in the original space then it will be linearly separable in the new space. The implication is that any valid similarity function can be used to map the data into a new space and then a standard linear separator algorithm can be used for learning.

3 Our Algorithm

Suppose $\mathbb{K}(x, y)$ is our similarity function and the examples have dimension k

We will create the mapping $\Phi(x) : \mathbb{R}^k \to \mathbb{R}^{k+d}$ in the following manner:

1. Draw d examples $\{x_1, x_2, \ldots, x_d\}$ uniformly at random from the dataset.
2. For each example x compute the mapping $x \to \{x, \mathbb{K}(x, x_1), \mathbb{K}(x, x_2), \ldots, \mathbb{K}(x, x_d)\}$
3. Run a linear separator algorithm such as Winnow on the expanded hypothesis space. Winnow is particularly suitable as it handles a large number of features very well.

Although the mapping is very simple, in the next section we will see that it can be quite effective in practice.

3.1 Choosing a Good Similarity Function

Defining a suitable similarity function was a major focus of our work. We called the procedure we ended up with *Ranked Similarity* and it is defined as follows:

1. Compute the similarity as before.

2. For each example x find the example that it is most similar to and assign it a similarity score of 1, find the next most similar example and assign it a similarity score of $(1 - \frac{2}{n-1})$, find the next one and assign it a score of $(1 - \frac{2}{n-1} \cdot 2)$ and so on until the least similar example has similarity score $(1 - \frac{2}{n-1} \cdot (n-1))$. At the end, the most similar example will have a similarity of $+1$, the least similar example will have a similarity of -1, with values spread linearly in between.

We found that this procedure was quite effective. The reasoning behind this procedure is explained in the full version of our paper.

3.2 Results

In Table 1 below, we present the results of our algorithm on a range of UCI datasets. In this table, n is the total number of data points, d is the dimension of the space, and nl is the number of labeled examples.We highlight all performances within 5% of the best for each dataset in bold.

Dataset	n	d	nl	Winnow	SVM	NN	SIM	Winnow+SIM
Congress	435	16	100	**93.79**	**94.93**	**90.8**	**90.90**	**92.24**
Webmaster	582	1406	100	**81.97**	71.78	72.5	69.90	**81.20**
Credit	653	46	100	**78.50**	55.52	61.5	59.10	**77.36**
Wisc	683	89	100	**95.03**	**94.51**	**95.3**	**93.65**	**94.49**
Digit1	1500	241	100	73.26	88.79	**94.0**	**94.21**	**91.31**
USPS	1500	241	100	71.85	74.21	**92.0**	86.72	**88.57**

Table 1: Performance of similarity functions compared with standard algorithms on some real datasets

We observe that on certain types of datasets such as the Webmaster dataset (a dataset of documents) a linear separator like Winnow performs particularly well, while standard Nearest Neighbor does not perform as well. But on other datasets such as USPS(a dataset comprised of images) Nearest Neighbor performs much better than any linear separator algorithm. The important thing to note is that the combination of Winnow plus the similarity features always manages to perform almost as well as the best available algorithm. For the UCI datasets we observe that the combination of the similarity features with the original features does significantly better than any other approach on its own. In particular it is never significantly worse than the best algorithm on any particular dataset.

4 Contributions

In this work we explored some ideas for learning using similarity functions and unlabaled data that have not previously appeared in the literature:-

1. Combining Similarity Based and “Native” features using Winnow.

2. Using unlabeled data to help construct a similarity function.

These ideas show promise for real applications and potential for theoretical development.

References

[Balcan and Blum(2006)] M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. *ICML06, 23rd International Conference on Machine Learning*, 2006.

[Shawe-Taylor et al.(1998)Shawe-Taylor, Bartlett, Williamson, and Anthony] John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory*, 44:1926–1940, 1998.