Selection of Image Features for Distribution-Map Classifiers

Tin Kam Ho

AT&T Bell Laboratories
600 Mountain Ave., Murray Hill, NJ 07974

ABSTRACT

A useful metric for pattern classification can be derived from a series of one-dimensional maps of empirical class-conditional distributions. To achieve maximum classification accuracy, the distributions need to be projected into subspaces where they are locally dense and the classes are at least partially separable. We describe a method where useful projections are obtained automatically by a sequence of cluster and linear discriminant analyses. Each projection can be considered as a feature that contributes to the elimination of unresolved ambiguities. We discuss conditions under which the method can achieve maximum accuracy. In an experiment of applying the method to machine-printed character images, we show that the method yields a classifier that has very low error rate, and can be improved with additional training data.

1 INTRODUCTION

Recently[4] we proposed a method for constructing a metric for pattern classification. The metric is defined on a series of mappings from the feature space to one-dimensional ranges. It is observed that classification using this metric results in very low error rate. The mappings in that study were designed by exploiting heuristics in the domain. For the classifier to be applicable to other domains, a general method is needed to derive necessary mappings automatically. In this paper we gives a method for this purpose.

We begin with a review on the principles of the metric and the classifier. Then we discuss requirements of the mappings and conditions under which a particular family of mappings will be useful. The focus of the paper will be on describing a method for deriving linear mappings that is applicable to a wide class of problems. The effectiveness of the method will be demonstrated through an experiment on the recognition of machine-printed characters.

2 DISTRIBUTION-MAP BASED CLASSIFICATION

One of the challenges in pattern recognition is to find faithful but concise descriptions of the class-conditional distributions of the samples. The samples are typically given in a high-dimensional feature space, and they often spread over regions of irregular geometrical shapes; both add to the difficulty of finding a compact expression. In fact, the design of classifiers will be easy if the distributions can be explicitly described in a compressed form.

These have motivated the design of the distribution-map based classifier recently proposed.[4] The classifier uses a metric for similarity that is defined on mappings of class-conditional distributions onto lower dimensional spaces. For complex distributions a series of mappings is needed. Each mapping projects the multi-dimensional distributions into a one-dimensional space, with discretized values in a bounded range. Each value in the range is marked if any sample of a class is projected onto that value. The marked values for each class constitute a "weak model" of that class in that particular one-dimensional space. The collection of such weak models gives a concise map of the distributions. For an arbitrary test sample, the metric counts the number of weak models of each class that do not include that sample. The metric is zero when the sample fits in all weak models of a particular class (a perfect match). Larger values of the metric indicate discrepancies of the sample from the models of the corresponding classes. Figure 1 shows the distribution maps of points of two classes and the projected values of a single sample.

There is no restriction on the form of the projections, given that they satisfy several criteria. First, the value range has to be bounded and discretized, and has a suitable cardinality. The cardinality (number of discrete values) needed depends on the number of available training samples. It has to be small enough so that training samples limited in quantity (say, a few hundreds) can form a dense model. It has to be large enough to reveal differences between samples of different classes. For this reason, binary values are usually not sufficient to differentiate the classes. As an example, in a previous experiment,[4] we found that a cardinality of 25 for each projection was useful with 800 training samples for each class.

A second criterion is that the projections should be different from each other. A projection need not be strictly orthogonal to all others. It may add to the discriminating power of the classifier as long as it is not identical to others. In principle, a projection does not introduce classification errors. The worst case is when distributions of all classes are projected onto identical values, so that the projection does not introduce any difference in the values of the metric.

By the definition of the metric, the classifier is more prone to indecisions due to ambiguities than to errors. A sample has a unique classification when the metric is minimized for one single class. This occurs when there is at least one value range in which the sample is outside all but one weak model. We now describe a method to obtain projections such that this condition is satisfied as far as possible. Following the convention in image recognition, we will call each of these projections a *feature* of the sample.



The projected values of one single sample.



The projections of samples of two classes.

Figure 1: Distribution maps of two classes and projections of a single sample. Indices of the projections are arranged along the horizontal axis, and the ranges of values are given along the vertical axis.

3 EFFICIENCY OF THE FEATURES

Given a fixed set of training samples of n classes, we need a set of projections so that each sample of each class is separated from samples of all other classes by at least one projection. We assume that there is no intrinsic ambiguities in the samples, i.e., no two classes share an identical sample.

A set of projections can be obtained trivially as follows: pick an arbitrary metric in the feature space, say, the Euclidean distance $d(x, y)$ between two samples x and y. For each sample x, compute $d(x, y)$ between x and all other samples $y(y \neq x)$. Set the value range to be $[0, max(d(x, y))]$. Then we have a mapping from the feature space onto this value range, which is simply the distance from this particular sample. Do this for all samples, and each sample will then be separated from all others by the distance measured from itself.

The projections thus obtained guarantee the discrimination between training samples of different classes, but they are often not satisfactory to be used in the distribution-map based classifier. The projections are too specific to particular samples, and too many of them are needed. For this reason we call these projections *inefficient*. A constant scaling factor may be used to compress the value ranges, so that the projected value remains invariant for samples in the immediate neighborhood of the sample from which

distance is measured. But this remedy may not work in application domains where samples are far apart in a large space, or when they are at almost the same distance from each other.

What we need are projections that better exploit local cluster characteristics of the samples, i.e., projections that can separate a larger set of samples from others. We will now focus on a condition on the class-conditional distributions that would allow discrimination by more efficient projections. We require that for a given problem in a d-dimensional feature space, for each sample x, there exists one or a number of parallel $(d-1)$-dimensional hyperplanes that can separate x from samples of all other classes. We call these classes *partially linearly separable*. Note that this condition is weaker than linear separability, which requires all samples of one class be separable from samples of all other classes by a single hyperplane. It occurs in practice that, in many applications with high-dimensional input, different classes rarely spread over nested regions, and the classes are often partially linearly separable. Figure 2 shows an example of two partially linearly separable classes.

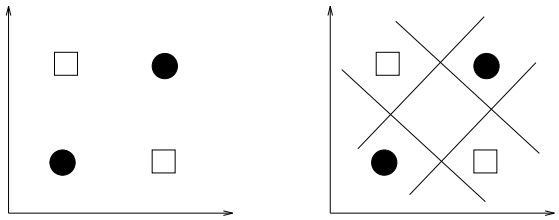


Figure 2: Two classes (c_1: black circles, c_2: white squares) that are not linearly separable but are partially linearly separable.

4 PROGRESSIVE ELIMINATION OF AMBIGUITIES

We now describe an algorithm that searches for projections that fully discriminate between partially linearly separable classes. The algorithm removes ambiguities progressively by introducing additional features that are targeted towards separating current ambiguous samples.

The algorithm is applied to each class in turn. Suppose we start with a particular class c. At the beginning, all training samples are first assigned to two sets S_1 and S_2, where S_1 consists of all samples from class c, and S_2 consists of samples from all other classes. The sample mean m_i of each set S_i $(i = 1, 2)$ are computed. A line $\overline{m_1 m_2}$ is drawn passing through m_1 and m_2. All samples are then projected onto this line. The range of the projection is then divided evenly into a fixed number of segments. A segment is marked for a class if there is any sample of that class projected onto that segment. A distribution map of the classes along this line is thus obtained.

If there is no segment marked for classes from both sets, then we have obtained a discriminating feature for class c. Otherwise, S_1 is pruned and only those samples overlapping with S_2 are retained (call the pruned set S_1'). The procedure is then repeated using S_1' and S_2. In case that all samples in S_1 overlap with those from S_2, S_1 is split into two halves and the same procedure is applied to each half. This continues until either S_1 is empty, or when it is impossible to separate S_1 and S_2 by any further projections.

Notice that when each new projection is made, some samples that are previously removed from S_1 may overlap with S_2 when they are projected onto the new line. But there is no need to backtrack to those samples because there has already been one projection on which they can be separated. After S_1 is finished, we put class c into S_2 and take another unprocessed class from S_2 to S_1, and the algorithm repeats itself until all classes are processed.

In this procedure, the directions of the projections are determined by the geometry of the spread of the classes, over which we have little control. An improvement can be made by introducing an optimization procedure to choose projections that lead to as few ambiguities as possible. The fixed increment perceptron training algorithm can be used in this context to find a hyperplane between S_1 and S_2, so that the number of samples falling on the opposite side is minimized. The only change from the previously described procedure is to replace the line $\overline{m_1 m_2}$ by the line perpendicular to the resultant hyperplane.

5 ADAPTATION TO UNSEEN SAMPLES

So far we have been concerned with separating all samples in a given training set. Given a fixed training set of a small size, this is easily achievable. However, the usefulness of the classifier is determined by its accuracy on unseen samples. In this section we discuss ways to adapt the classifier to new samples.

The quantization of the value ranges of the features provides limited invariance of classification among samples projected onto the same line segment. By the definition of the metric used by the classifier, an error occurs when a test sample has a feature value that has not previously occurred among training samples of that class. To prevent this from happening, the training set must be large enough relative to the range and quantization of the feature values, so that the projections can be fully saturated. Yet too many training samples could also lead to the problem of over-saturation – in an extreme case where all class-conditional distributions have infinite nonzero tails, any projection will eventually have all values marked for all classes, and when this happens no discrimination will be possible. In the previous discussions we have excluded these cases by assuming that each sample have a unique classification. But in practice this often cannot be guaranteed.

It is therefore difficult to predict the quantity of training samples needed. An adaptive procedure is preferred, so that the classifier can improve itself by using additional training samples. It is possible that new training samples be projected to values that are marked for other classes but unmarked for their classes,

and thus introduce new ambiguities. The classifier will need new features to resolve such ambiguities. If we keep adding in new training samples and find no further need to include new features, the classifier can then be considered stable. Errors will also be unlikely since newer samples are then less likely to turn on values unmarked for their classes.

The way new features are found is similar to the selection of other features. We first project the new training samples using existing mappings, and identify those that overlap with other classes. Then we look for hyperplanes that separate those samples from other classes. Older training samples of other classes also need to be re-examined for overlaps with the new samples.

6 EXPERIMENTAL RESULTS

We tested the procedure in an experiment on a constrained problem in optical character recognition. We chose the problem of distinguishing between images of the symbols 'c' and 'e' in the Adobe Times Roman typeface, with noise introduced by a parameterized model of document image defects.[1] The choice was motivated mainly by the practical importance of the problem and the unambiguous shapes of the ideal prototypes. The use of the defect model gives us a way to precisely define the scope of the problem, and also an infinite source of data for training and testing. The values of the defect model parameters are identical to those used in a previous experiment,[5] and will be omitted here for brevity.

Using the defect model and the ideal shape prototypes, we generated 500,000 training samples that are divided into 59 sets. There are 1,000 (500 'c's and 500 'e's) samples in each of the first ten training sets, and 10,000 (5,000 'c's and 5,000 'e's) in the subsequent ones. The test set consists of 50,000 'c's and 50,000 'e's (100,000 samples in total). The same test set is used to measure the error rate throughout the experiment, regardless of changes in the training set and the features. Each sample image is binarized, size-normalized to 48×48 pixels. The binary values of the pixels are used as input.

At each iteration, linear projections were found using the fixed increment perceptron training algorithm. For each projection, the range of values was divided into 25 segments. Once sufficient features were found to fully discriminate a given training set, the next training set was added and the algorithm searched for new features to resolve new ambiguities. Because of difficulties in implementation, we considered ambiguities in the new training set only and did not backtrack and re-examine previous training samples.

When the algorithm finished with a training set, we tested the accuracy of the classifier with the fixed test set. Since we have an infinite source of training data, in principle this could continue indefinitely. However, seeing no new pattern on the results, we terminated the experiment after 500,000 training samples were used. Table 1 shows the changes in classification accuracy and the number of features added with each new training set.

Table 1: Changes in classifier accuracy and size

training set added	total #ims. for training	#ims. correct	% correct	#ims. error	% error	#ims. reject	% reject	#new features	total # features
1	1000	99749	99.75	52	0.05	199	0.20	1	1
2	2000	99828	99.83	56	0.06	116	0.12	2	3
3	3000	99848	99.85	60	0.06	92	0.09	1	4
4	4000	99836	99.84	68	0.07	96	0.10	2	6
5	5000	99718	99.72	59	0.06	223	0.22	2	8
6	6000	99792	99.79	77	0.08	131	0.13	2	10
7	7000	99775	99.78	60	0.06	165	0.17	2	12
8	8000	99828	99.83	60	0.06	112	0.11	2	14
9	9000	99759	99.76	50	0.05	191	0.19	0	14
10	10000	99691	99.69	56	0.06	253	0.25	1	15
11	20000	99841	99.84	66	0.07	93	0.09	2	17
12	30000	99932	99.93	23	0.02	45	0.04	2	19
13	40000	99953	99.95	22	0.02	25	0.03	2	21
14	50000	99931	99.93	22	0.02	47	0.05	3	24
15	60000	99890	99.89	21	0.02	89	0.09	2	26
16	70000	99867	99.87	23	0.02	110	0.11	3	29
17	80000	99941	99.94	28	0.03	31	0.03	2	31
18	90000	99913	99.91	25	0.03	62	0.06	3	34
19	100000	99860	99.86	20	0.02	120	0.12	3	37
20	110000	99903	99.90	37	0.04	60	0.06	3	40
21	120000	99872	99.87	16	0.02	112	0.11	2	42
22	130000	99900	99.90	25	0.03	75	0.07	2	44
23	140000	99893	99.89	14	0.01	93	0.09	3	47
24	150000	99895	99.89	22	0.02	83	0.08	3	50
25	160000	99909	99.91	16	0.02	75	0.07	3	53
26	170000	99890	99.89	29	0.03	81	0.08	2	55
27	180000	99918	99.92	16	0.02	66	0.07	3	58
28	190000	99905	99.91	21	0.02	74	0.07	3	61
29	200000	99868	99.87	16	0.02	116	0.12	3	64
30	210000	99909	99.91	38	0.04	53	0.05	3	67
31	220000	99905	99.91	27	0.03	68	0.07	3	70
32	230000	99893	99.89	20	0.02	87	0.09	2	72
33	240000	99886	99.89	18	0.02	96	0.10	3	75
34	250000	99888	99.89	26	0.03	86	0.09	3	78
35	260000	99909	99.91	26	0.03	65	0.07	2	80
36	270000	99884	99.88	27	0.03	89	0.09	3	83
37	280000	99854	99.85	20	0.02	126	0.13	3	86
38	290000	99924	99.92	35	0.04	41	0.04	2	88
39	300000	99874	99.87	23	0.02	103	0.10	2	90
40	310000	99879	99.88	27	0.03	94	0.09	4	94
41	320000	99888	99.89	23	0.02	89	0.09	2	96
42	330000	99898	99.90	16	0.02	86	0.09	3	99
43	340000	99893	99.89	16	0.02	91	0.09	2	101
44	350000	99881	99.88	37	0.04	82	0.08	3	104
45	360000	99910	99.91	25	0.03	65	0.07	2	106
46	370000	99881	99.88	16	0.02	103	0.10	2	108
47	380000	99873	99.87	26	0.03	101	0.10	2	110
48	390000	99834	99.83	16	0.02	150	0.15	4	114
49	400000	99877	99.88	15	0.01	108	0.11	2	116
50	410000	99883	99.88	17	0.02	100	0.10	2	118
51	420000	99845	99.84	23	0.02	132	0.13	3	121
52	430000	99890	99.89	21	0.02	89	0.09	2	123
53	440000	99839	99.84	23	0.02	138	0.14	3	126
54	450000	99864	99.86	15	0.01	121	0.12	3	129
55	460000	99871	99.87	41	0.04	88	0.09	3	132
56	470000	99872	99.87	25	0.03	103	0.10	2	134
57	480000	99892	99.89	21	0.02	87	0.09	3	137
58	490000	99906	99.91	25	0.03	69	0.07	2	139
59	500000	99919	99.92	33	0.03	48	0.05	2	141

From Table 1 it is not difficult to see that the accuracy of the classifier became stable from early on (when 100,000 training samples were used). This is more obvious from the graphs shown in Figure 3.

The results suggest that the initially derived features contribute most to classification accuracy. We can measure the efficiency of each feature by the number (or %) of samples separated from other classes by using that feature. As an example, we examined the 139 features obtained before training set 59 was added. Figure 4 shows the numbers of samples distinguished by each of those 139 features, starting from the features derived using training set 1. This graph illustrates the ability of the method in selecting the most useful features first. Since the features are ordered by their efficiency, the sequence can be pruned according to requirements on accuracy and availability of resources.

It is unclear what has caused the persisting errors and rejects. Our conjecture is that the class-conditional distributions have nonzero and overlapping tails, so that there are certain intrinsic ambiguities in the data – relative to the quantization of values we have adopted.

7 ALTERNATIVE PROCEDURES

A few other trials have been carried out using several alternative procedures. The trivial method of obtaining mappings by Euclidean distance, as expected, produced very inefficient features. The method of projection on the line connecting cluster centroids also yielded many inefficient features, with low error rates but high reject rates. Using the same perceptron method with value ranges divided into 50 segments instead of 25 resulted in slower improvements in accuracy, though after 50,000 training samples were used the differences were no longer obvious.

The method of *principal components* (Karhunen-Loéve expansion)[2] was considered for finding useful directions for linear projections. However, maximization of the variability of the entire data set does not contribute much to the *discrimination* between classes. For a similar reason, we did not find the method of *projection pursuit*[3] in exploratory data analysis useful in this context.

8 CONCLUSIONS

We have shown a way to construct a classifier automatically – both the features and the metric were derived with minimum heuristics. Moreover, the classifier can be improved with additional training data to higher accuracy.

In this study we have concentrated on linear mappings as features. Future studies may focus on investigating other families of features, especially those that can be used under weaker conditions on the class-conditional distributions.

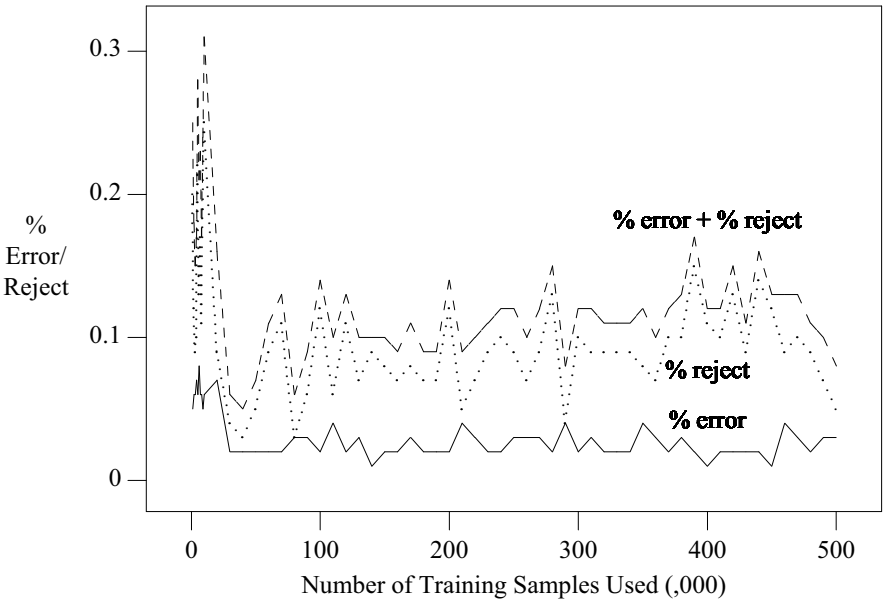


Figure 3: Changes in error and reject rates as training samples were added.

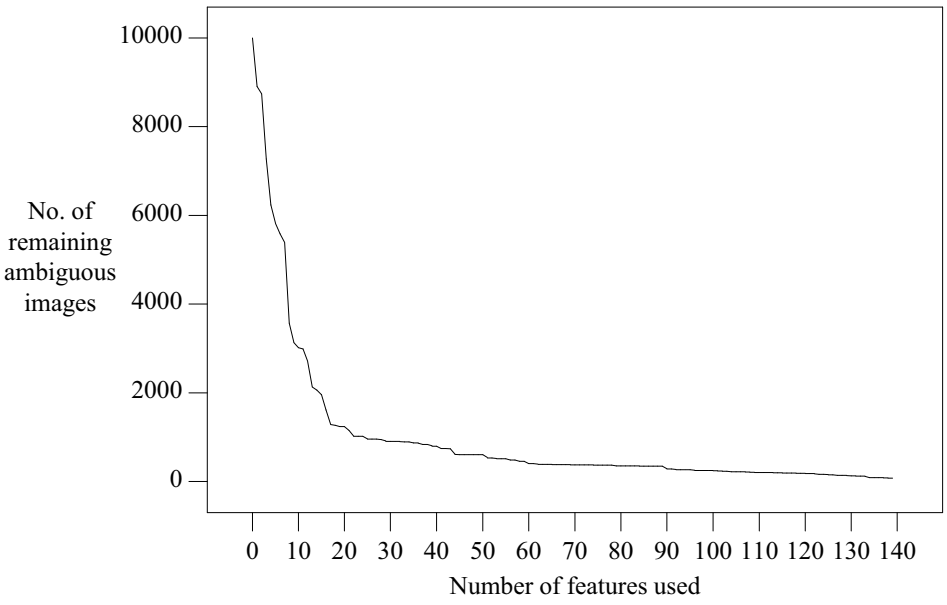


Figure 4: Efficiency of individual features in distinguishing ambiguous images.

9 ACKNOWLEDGEMENTS

The author thanks Henry Baird, Dave Ittner, and George Nagy for many useful discussions.

10 REFERENCES

[1] H.S. Baird, Document Image Defect Models, in H.S. Baird, H. Bunke, K. Yamamoto (Eds.), *Structured Document Image Analysis*, Springer-Verlag, 1992.

[2] R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Addison-Wesley, New York, 1973.

[3] J.H. Friedman, J.W. Tukey, A Projection Pursuit Algorithm for Exploratory Data Analysis, *IEEE Transactions on Computers*, **c-23**, 9, September 1974, 881-890.

[4] T.K. Ho, H.S. Baird, Perfect Metrics, *Proceedings of the Second International Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, October 20-22, 1993, 593-597.

[5] T.K. Ho, H.S. Baird, Asymptotic Accuracy of Two-Class Discrimination, *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, April 11-13, 1994, pp.275-288.