An Empirical Model for Saturation and Capacity in Classifier Spaces *

R. B. Fisher
University of Edinburgh - rbf@inf.ed.ac.uk

Abstract

When assessing reported classification results based on selection of members from a database (e.g. a face database), one would like to know what is an achievable classification rate, given the noise level, dimensionality of the feature set and number of classes in the database. As best we can tell, no general results exist for this question, although many classification rates appear in different papers. This paper presents an empirical formula for MAP classification that links the number of discriminable classes to the error rate, dimensionality of the feature data and the feature noise level.

Keywords: information retrieval, vision, decision theory, classifier performance

1. Introduction

Consider a typical classification problem: "Is face F one of those in a known database?". A commonly encountered research methodology used to validate the recall performance of a given algorithm is to probe the database with a collection of known targets and then report the percentage P of retrievals with the correct candidate found, or within the top T candidates (*e.g.* for faces [4, 7, 8] and for 3D models [2]). This seems like a good methodology, but we are then left asking: "Is P a good percentage?" (*i.e.* is it approaching the best possible percentage?)

A second related problem is: how long can the good performance continue as additional new classes/faces are added? It is obvious that if there are only a few classes/faces, then it is probably easy to discriminate between them. But, as the number of classes/faces becomes large, then performance is certain to drop dramatically as the classifier space becomes "saturated". What is the threshold? Understanding these issues is necessary to extrapolate from reported results to likely real-world performance.

These issues arise in a wide range of pattern recognition and information retrieval problems [1, 6].

As best we can tell, no general answers have been published for these and related questions. **This paper presents and justifies an empirical formula for MAP classification that links the number of discriminable classes to the error rate, dimensionality of the feature data and the feature noise level.** The closest modern results focus on machine learning

*Many thanks to Scott Blunsden, Toby Breckon, Miguel Cazorla, Josef Kittler, Tim Lukins, David Tweed and Chris Williams for advice on this problem.

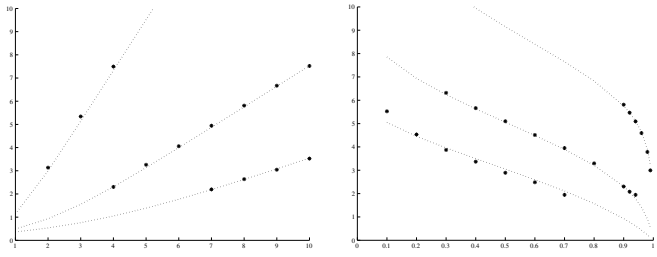

Figure 1. Observed (asterisks) and predicted (dotted line) log of the maximum number of discriminable classes $\log(N)$ (vertical) *versus* LEFT: data space dimensionality D (horizontal) for noise level σ = 0.1 (top curve), 0.4 (middle curve), 0.7 (bottom curve), for the top classification only $T = 1$, and classification rate $p = 0.9$. RIGHT: classification rate p (horizontal) for data space dimensionality D = 2 (bottom curve), 4 (middle curve), 8 (top curve), for the top classification only $T = 1$ and noise level $\sigma = 0.4$.

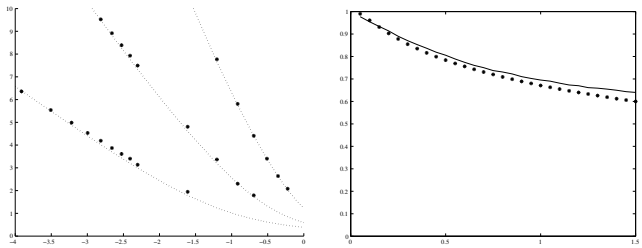

Figure 2. LEFT: Observed (asterisks) and predicted (dotted line) log of the maximum number of discriminable classes $\log(N)$ (vertical) *versus* log of noise level $log(\sigma)$ (horizontal) for data space dimensionality D = 2 (bottom curve), 4 (middle curve), 8 (top curve), for the top classification only $T = 1$ and classification rate $p = 0.9$. RIGHT: Predicted classification rate (p) versus noise level σ for the case data space dimensionality $D = 1$, number of classes $N = 2$, for the top classification only $T = 1$. The solid line is the known true result from using the erfc() function and the asterisks show the predictions from formula (3).

convergence issues: *i.e.* how much capacity (parameter dimensionality) is needed to allow efficient and correct classifier generalization (*e.g.* [3]) given a set of training data, with the underlying assumption of perfect classification. Here, we raise questions about controlling the best possible classification rates when errors are impossible to avoid, and also allowing multiple top choices, *i.e.* performance is correct if the true classification is in the top T retrievals.

An indirect inspiration for this problem is via sphere packing, where Kepler conjectured and Hales proved [5] that the maximum density of packing was $\frac{\pi}{2\sqrt{2}} \doteq 74\%$. One could treat the feature space around each class prototype as a sphere. Thus, by setting the classification error rate and knowing the class error distribution we can compute a sphere radius that contains the required percentage of the classifications. This sphere packing approach is inadequate, in that: 1) it assumes uniform placement of the classes, which is unrealistic, 2) the true (linear?) classification boundaries mean that there could be some correctly classified features lying outside the spheres, and 3) this approach only allows for the top class, whereas we want to consider the top T classes.

This criteria of considering a success if the true class is in the the top T selected classes adds much complexity to the analysis, as now success occurs over all partitions of the feature space where the true class is one of the top T labels. This is more complex than simply the portion of the feature space closest (in a probabilistic sense) to a given class prototype.

The empirical formula given below gives answers for the questions raised above; at least as a guideline (because of several simplifying assumptions made). Further, it is really only an upper limit on performance because the analysis makes assumptions of independent and randomly distributed data, whereas real data is not as ideal.

2. Problem Model

Assume a database of dimension D (*i.e.* D-dimensional feature vectors) with N different classes/objects. We formulate the above questions in a more general form as:

1. What is the probability that an unknown sample $\vec{x}$ with known noise $\vec{\sigma}$ is classified correctly (*i.e.* the correct class is the class with the highest probability of explaining the observed sample)?
2. What is the probability that an unknown sample $\vec{x}$ with known noise $\vec{\sigma}$ has the true class within the T most probable class prototypes?
3. What is the maximum number of classes N usable while still maintaining an average probability p of correct classification?
4. What is the minimum dimension D of the independent feature space needed to classify N objects with a probability of correct classification rate of at least p?

We formulate the problem as follows: assume N classes. Each class is represented by feature vectors $\vec{x}$ of dimension D. Class i has a probability distribution $p_i(\vec{x}, \vec{\mu}_i, \vec{\epsilon}_i)$ centered about mean point $\vec{\mu}_i$ with distribution parameters $\vec{\epsilon}_i$. Later we will assume that this distribution is Gaussian. Assume that the class centers $\vec{\mu}_i$ are also distributed with a probability distribution $p_C(\vec{\mu}_i, \vec{\eta}, \vec{\epsilon}_C)$ with distribution parameters $\vec{\epsilon}_C$, centered about $\vec{\eta}$ (the coordinate system origin in the case of the experiments below), which we will also model as Gaussian later. Assume that the *a priori* probability of each class is a_i, with $\sum a_i = 1$.

Assume a classifier that assigns the most likely class to an observation $\vec{x}$ (*e.g.* a Bayes' classifier). Define $g_i(\vec{x}, T) = 1$ if the likelihood that observation $\vec{x}$ belongs to class i is higher than at least $N - T$ other class likelihoods. Otherwise

$g_i(\vec{x}, T) = 0$. $g_i(\vec{x}, T)$ partitions the feature space $\mathcal{R}^D$ into regions where $\vec{x}$ is classified as class i ($g_i(\vec{x}, T) = 1$) or classified as another class ($g_i(\vec{x}, T) = 0$). If $T = 1$, then the partition gives regions where the likelihood of a single class is higher than all others, like a Voronoi diagram.

We simplify the notation for the N class distribution centers $\{\vec{\mu}_i\}$ by creating a new $N \times D$ vector $\vec{v}$ by concatenating all of the class mean vectors $\vec{\mu}_i$. Thus $\vec{v} \in \mathcal{R}^{N \times D}$. We then create a new probability distribution $p_V(\vec{v}, \vec{\eta}, \vec{\epsilon}_V)$ modelling the combined distribution of the set of classes given all of the original distributions.

With this model, given a distribution of the class distribution centers $\vec{\mu}_i$, the average probability $f(\vec{v}, T)$ of correct classification within the top T classes over all classes i for a given sample $\vec{v}$ of the class center distribution is:

$$f(\vec{v}, T) = \sum_{i=1}^{N} a_i \int_{-\infty}^{+\infty} p_i(\vec{x}, \vec{\mu}_i, \vec{\epsilon}_i) g_i(\vec{x}, T) d\vec{x} \qquad (1)$$

The inner integral computes the average probability that class i is in the top T selections. The sum then weights by the *a priori* class probabilities. Finally, we can compute the average probability of correct classifications over all distribution of class centers $\vec{v}$ by:

$$p_t(T) = \int_{-\infty}^{+\infty} p_v(\vec{v}, \vec{\eta}, \vec{\epsilon}_V) f(\vec{v}, T) d\vec{v} \qquad (2)$$

Thus, we have a theoretical formula that links the number of classes N to the average classification error rate $1 - p_t(T)$, the size of the acceptability set T, the dimensionality D, the distribution of class centers $p_v(\vec{v}, \vec{\eta}, \vec{\epsilon}_V)$ and the distribution of observations about those centers $p_i(\vec{x}, \vec{\mu}_i, \vec{\epsilon}_i)$. So, in theory, to compute the maximum number of usable classes, we just need to solve these formulas for N, given the other information. As solving this symbolically seems to be impossible, we present an empirical solution in the next section.

3. Claim

We propose a solution for many classification problems:

- The number of measured features (transformed to be independent) is D (*i.e.* the dimension of a sample $\vec{x}$).
- The distribution of the observed feature values for each class i of objects is a multi-dimensional Gaussian, with mean $\vec{\mu}_i$ scaled so that the standard deviation of feature j is σ_{ij}. (Recall independence of transformed features from the previous assumption.) To simplify the problem, we assume that $\sigma_{ij} = \sigma$, but the analysis below could use $\sigma = min_{ij}(\sigma_{ij})$ to get an upper bound on N.
- The distribution of the mean vectors $\vec{\mu}_i$ of each of the different classes is a multi-dimensional Gaussian centered at the origin, where $\mathbf{A}$ is the class distribution covariance matrix and all classes are *a priori* equally likely ($a_i = a_j$ in Eqn (2)).
- The class selection criteria is the maximum *a posteriori* probability $pr(i \mid \vec{x})$ of the sample with feature vector $\vec{x}$ having come from class i. This is a standard probabilistic classification method. (Other classifiers such as k-NN or a rule/decision tree algorithm could also be analyzed, but we consider only the MAP classifier in the analysis presented here.)

Assume that a correct classification is declared if the true class of an observed sample $\vec{x}$ is within the T most probable classes. Then, the following *empirical* formula predicts the maximum number of classes N potentially discriminable with average classification error rate p, assuming the feature noise level σ over D observed properties.

$$N = T + a(p, T)\sqrt{det(\mathbf{A})} \left(\frac{b(p, T)}{\sigma / (1 + 1.18\sigma)} \right)^{(c(p,T)D + d(p,T))} \qquad (3)$$

where

$$\begin{aligned} a(p, T) &= exp(2.15 + 0.50T + p(-1.20 - 0.45T) + 1.06(1/T)\log(1 - p)) \\ b(p, T) &= 0.46 - 0.03T + p(0.12 - 0.03T) - (0.15 - 0.02T)\log(p) \\ c(p, T) &= 1.14 + 0.18T + p(0.06 - 0.22T) \end{aligned}$$

$$d(p,T) = 0.04 - 0.15T + p(-0.56 + 0.30T) - \qquad (0.14 + 0.02T)\log(1-p)$$

Because of the complexity of the multiple overlapping classes and the need to account for multiple top classes (T), it seems unlikely that a closed form symbolic solution for this problem is possible. Thus, we have taken an engineering approach: 1) observe performance under experimental conditions and 2) fit numerical models to the observed results. Formula (3) is the result. The experimental data is based on Monte-Carlo simulations as described in the next section. The basic structure of the equation was chosen after observing the nearly linear sections of the data observed when $log(N)$ is plotted *versus* D, p and $log(\sigma)$ (See Figures 1 - 2). Additional function terms were added to model the non-linear effects at the extreme values for D, p and σ. Numerical parameters in the formulas were found by Matlab-based optimized curve fitting to the simulation results.

As well as fitting the simulation data, formula (3) also has the desired qualitative behavior, as summarized in the table here (use class distribution $det(\mathbf{A}) = 1$, noise level $\sigma = 0.1$, desired classification rate $p = 0.9$ and data dimensionality $D = 8$ as the defaults while exploring other variables):

Limit	True N	Predicted N
$\sigma \to 0$	∞	∞
$\sigma \to \infty$	T	T
$D \to 1$	finite $\geq T$	finite $\geq T$
$D \to \infty$	∞	∞
$p \to 0$	∞	∞
$p \to 1$	T	T
$T \to 1$	finite ≥ 1	finite ≥ 1
$T \to \infty$	∞	∞

4. Evidence for Formula

4.1. Fit to simulation data

The claim made in Section 3 was validated empirically by Monte Carlo simulation, where the simulation data was generated based on the problem model presented at the start of Section 3. For a given set of parameters, N class centers $\vec{\mu}_i$ were chosen randomly from a normal $(\vec{0}, I)$ distribution. (*i.e.*, the class distribution covariance matrix $\mathbf{A}$ is the identity.) Random samples $\vec{x}$ of each class were selected from a normal $(\vec{\mu}_i, \sigma^2 I)$ distribution. The most likely class (in this case the closest class prototype) was then identified for each sample $\vec{x}$ and the percentage of these that were correct is the classification rate. The experiments were repeated a given number of trials ("Trials") and the classification rates averaged. These became the 3000+ data points that the model of eqn (3) was fitted to, a small subset of which are shown as the asterisks in Figure 1-2. We computed the probability of correct classification p for the set of dimensions D and noise standard deviations σ given in the table below. The investigated number of classes N was varied from 1 to max(N) in steps of step(N). Some specific trials were run to fill in a few additional data points in the plots. About 1000 CPU hours (in Matlab) were used to generate the data.

D	σ	max(N)	step(N)	Trials
1-2	0.01-0.09	1000	1	100
3-4	0.01-0.05	50000	500	10
3-4	0.06-0.09	20000	50	10
5-10	0.01-0.09	50000	1000	10
1-3	0.1-1.0	500	1	500
4-8	0.1-0.5	1000	2	500
4-8	0.6-1.0	500	1	500
9-10	0.1-0.5	3000	10	100
9-10	0.6-1.0	1000	2	500
20-50	0.1-1.0	50000	2000	10

Figures 1 - 2 show the observed $\log(N)$ values *versus* D, σ and p for different T, p, σ and D. The data values from the Monte Carlo simulation are plotted with asterisks. Superimposed in dotted lines are the values predicted by formula (3). Each plot shows several curves for different values of other independent variables. As can be seen, the predictions are very good, although deviate slightly at the extremes of the explored parameter range.

Figure 1 shows the log of the maximum number of classes (N) *versus* the dimensions of the feature space D for three noise levels σ and other parameters fixed. This figure is for when only the most likely class is considered ($T = 1$). The curves clearly fit the data points well, and also have the correct quantitative behavior: more classes can be accommodated if the data dimensionality (D) increases or if the noise level decreases (σ). Figures 1 (right) and 2 show the log of the maximum number of classes (N) *versus* the desired correct classification rate p and log of the data noise level σ for three different data space dimensionalities (D). Both plots again show good fit of the Monte Carlo (asterisks) to the predictions (dotted line) from formula (3).

The figures show that formula (3) predicts the observed values well. The numerical approximation breaks down slightly at the extremes of the parameter ranges. Mostly, the model deviations are insignificant, since the likely "working range" of the model is $p > 0.5$, small σ and medium D. Similar results are observed for the cases of $T = 2, 3$ but these are not shown due to paper length.

Suppose we consider the case with only 2 classes ($N = 2$) in one dimension ($D = 1$) and we only want the classification error for the top class ($T = 1$). In this case, we can numerically solve formula (3) for p as a function of σ, and also relate it to the expected true error (by using the erfc function). Figure 2 (right) shows the true error as a solid curve and the predictions as the asterisks. The two show good correspondence, with the predictions being about 4% smaller at $\sigma = 1.5$.

Over all of data, the average relative error ((observed N - predicted N)/observed N) is 1% for $p \leq 0.96$ and $n > 10$. The average relative absolute error (| observed N - predicted N |/observed N) is 11%. The worst relative error is at the extremes of the model, where $p \rightarrow 1.0$, D is big and σ is big.

4.2. Fit to character data

To test performance against real data sets, we used a subset of the NIST handwritten character dataset (`www.itl.nist.gov/iad/8`
`/defs/dbases.html`). 5000 samples each of either $N = 10$ numerical or $N = 36$ numerical and alphabetic characters were used. The original input data was a 32×32 binary image of a centered digit or character. The ground truth classes of each character is known. These images were then projected onto a D dimensional subspace using the D most significant eigenvectors found by PCA on the $5000N$ samples. The vector of coefficients from the projection became the data vectors $\vec{x}$ classified here. The class means $\vec{\mu}_i$ were estimated from these vectors. To use formula (3), we need an estimate of the in-class noise σ. This was estimated by

$$\sigma_i = \sqrt{\frac{trace(\mathbf{C}_i)}{trace(\mathbf{C}_g)}}$$

where $\mathbf{C}_i$ is the covariance matrix of class i and $\mathbf{C}_g$ is the covariance matrix of the whole dataset. This gives the average over all dimensions of the standard deviation scaled by the average for the whole dataset. The classifier selected the class i whose Mahalanobis distance to the class mean $\vec{\mu}_i$ is smallest. Because we are investigating maximum achievable performance rather than classifier learning rates, we used the same data for training and classification.

The table below shows the predicted ($p_{predicted}$) and actual classification rates ($p_{achieved}$) for this data, over either $N = 10$ or $N = 36$ classes and a variety of feature vector dimensions D and with the corresponding estimated noise levels σ. The classification rates p were estimated numerically from formula (3).

N	D	$\sigma_{estimated}$	$p_{predicted}$	$p_{achieved}$
10	1	0.498	0.409	0.326
10	2	0.603	0.544	0.471
10	3	0.610	0.667	0.668
10	4	0.701	0.709	0.670
10	5	0.723	0.772	0.723
10	10	0.761	0.946	0.870
10	15	0.800	0.987	0.902
10	30	0.844	0.999	0.920
36	1	0.580	0.064	0.087
36	2	0.708	0.207	0.152
36	3	0.677	0.364	0.272
36	4	0.699	0.473	0.383
36	5	0.742	0.545	0.420
36	10	0.783	0.837	0.651
36	15	0.808	0.955	0.736
36	30	0.850	0.999	0.791

The predictions for the hand-written digits are slight overestimates, but are close. The predictions for the full digit and letter database is less accurate; however, there is much similarity between some classes (*e.g.* {1, i, l (el)}, {0, o}), which reduces the observed classification accuracy by an estimated 8%. In addition, the distribution of the different classes was not Gaussian given the small number of classes. Estimating the noise level parameter σ is also imperfect, as the model assumes a hyperspherical distribution, whereas the real distributions will be different. Finally, the features used here are not as independent and effective as those specially designed for the character recognition process. Thus we can expect the experimentally observed classification rates to be lower than the predictions. Nonetheless, the approach gives useful numerical predictions of the maximum achievable performance, particularly for lower dimension feature vectors.

5. Discussion

Given formula (3), the answers to the questions in Sections 1 & 2 can be estimated trivially through directly solving for one of N, $det(\mathbf{A})$, σ or D as a function of the other parameters, and numerically if p or T are desired. The simulation and character set data both support the empirical model.

In developing this empirical model, we have assumed:

- **Independent features** - This is probably achievable in practice by PCA or ICA or kernel methods.
- **Properties have identical standard deviation** σ - This is partly achievable by scaling if all classes have the same distribution. Otherwise using the max and min of the σ_i allows estimation of bounds on N.
- **Properties have a Gaussian distribution** - This is questionable, but this probably means that formula (3) predicts larger Ns than for other distributions.
- **Each class has the same distribution** - This is also questionable, but may be reasonable if the processes that create variance are the same for all classes.
- **The distribution of class means is Gaussian** - This depends on the population being classified, but it can be a reasonable assumption.

Obviously, formula (3) has an empirical basis. It might be possible to derive a symbolic formula for $T = 1$ based on nearest neighbor overlap and small σ, assuming effectively no overlap at a distance. A symbolic model is unlikely for $T = 2, 3$ because of multiple overlapping distributions. Maybe a more precise empirical formula could be found by using more substantial simulations or better approximation functions.

We have made several simplifying assumptions that mean a user's data would need to be rescaled in order to make firm statistical predictions, but the character dataset results show that the approach is feasible. The results also show qualitatively how classifier spaces saturate, and how additional feature dimensions can quickly increase the number of discriminable classes.

In conclusion, in spite of its empirical basis, formula (3) gives usable results for predicting new or assessing published MAP classification rates. We are still investigating whether a symbolic model (partial for $T = 1$ or full) can be developed, based on ideas from communication channel theory.

References

[1] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, Wiley, 2001, Ch 2.

[2] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A Search Engine for 3D Models. ACM Transactions on Graphics, 22(1), 2003.

[3] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S.A. Solla. Capacity control in linear classifiers for pattern recognition. Proc. 11th Int. Conf. on Pattern Recognition, volume II, pages 385–388, 1992.

[4] M. J. Jones and P. Viola. Face Recognition Using Boosted Local Features. Mitsubishi Electric Research Laboratories Technical Report Number: TR2003-25. Date: April 2003.

[5] T. C. Hales. The Sphere Packing Problem. J. Comput. Appl. Math 44, 41-76, 1992.

[6] D. Lewis. Evaluating text categorization. Proc. Speech and Natural Language Workshop, pp. 312–318, Feb. 1991.

[7] S. Romdhani, V. Blanz, T. Vetter. Face identification by fitting a 3D morphable model using linear shape and texture error functions. Proc 7th Eur. Conf. on Comp. Vis., Part IV, pp 3–19, 2002.

[8] G. Shakhnarovich, J. W. Fisher, T. Darrell. Face recognition from long-term observations. Proc 7th Eur. Conf. on Comp. Vis., Part III, pp 851–865, 2002.