Optimal Data Partition for Semi-Automated Labeling

Daniel Lopresti
Lehigh University
Bethlehem, PA, 18015 USA
lopresti@cse.lehigh.edu

George Nagy
Rensselaer Polytechnic Institute,
Troy, NY, 12180 USA
nagy@ecse.rpi.edu

Abstract

In a pattern recognition sequence consisting of alternating steps of interactive labeling, classifier training, and automated labeling (e.g., CAVIAR systems), the choice of sample size at each step affects the overall amount of human interaction necessary to label all the samples correctly. The appropriate splits depend on the error rate of the classifier as a function of the size of the training set and, perhaps surprisingly, are independent of the relative costs of interactive correction and confirmation. We model such a system and report the sequence of optimal data partitions for a representative range of parameters.

1. Introduction

The following paradigm is the common denominator of some Computer Aided Visual Interactive Classification (*CAVIAR*) systems developed for diverse applications (Fig 1):

1. A sequence of N_{train} patterns is displayed on a graphic user interface and labeled by an operator.
2. A conventional classifier is trained on features extracted from the labeled patterns.
3. The classifier is run on a (usually larger) set of unlabeled patterns.
4. The newly labeled patterns are displayed for approval or correction by the operator.
5. Optionally, the corrected or approved labels and patterns are added to previous training set to retrain the classifier for the next batch of unknown patterns.

The cost of entering the label of a training pattern or of correcting a misclassified test pattern is higher than that of confirming a correct label (which typically requires only a single mouse click). It has been our experience that corrections take 5-6 times longer than confirmation for flowers or web table cells. The relative costs can be determined from a short interactive session. A ratio of 3 is appropriate for digit recognition due to the smaller number of classes.

We consider computational costs of interaction plus classification essentially constant and endeavor to minimize operator time. Multistage CAVIAR systems incur lower human interaction costs than the customary alternative of training on a small fixed partition and correcting every error at the end.

Given a fixed number of patterns to be labeled with a CAVIAR system, it is necessary to decide the appropriate balance between the size of the training set and the size of the interactively corrected test set. A large training set decreases the number of errors on the test set, but labeling it is expensive. A small training set may results in too many costly-to-correct errors on the test set.

We propose a model for determining the optimal size of the training set given a fixed number of patterns to be classified. We consider both two-stage (Steps 1-4 above) and *n*-stage (Steps 1-5) CAVIAR systems.

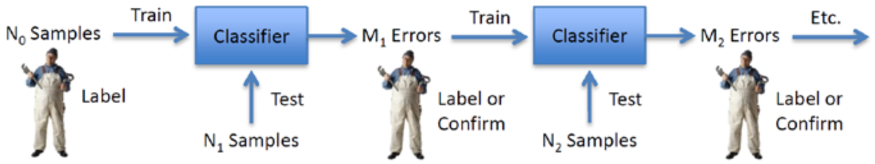

Figure 1. CAVIAR dataflow

Table I. Some CAVIAR systems

Objects	Sample Size	Label	Year	Reference
Printed documents	30,000	ASCII code	1971	[1]
Smudged print	28,650	Prototypes for classification	1998	[2]
Flowers	612	Species	2003	[3,4]
Calligraphy	13,351	GB-2312 code	2011	[5]
Web tables	30,795 (cells)	Cell type (header, stub, data, …)	2012	[6]
Faces	500	Name of person	ongoing	[7]
Election ballots	13,315	Candidate	″	[8]
Cervigrams	100	Abnormal tissues	″	[9]

We note that in some CAVIAR systems, the features, instead of the labels, of the misclassified patterns are corrected to retrain the classifier. Other systems include a *reject* class for which the operator must enter a new label rather than correct or confirm one. This is only an academic distinction if all the patterns are inspected because changing "?" to "Jones" takes about the same amount of effort as changing "Smith" to "Jones".

2. Prior work

CAVIAR systems have been developed for printed documents [1, 2], flowers [3, 4], calligraphy [5], and web tables [6]. Similar systems are being developed for faces [7], election ballots [8] and cervigrams [9]. Table I displays some characteristics of these systems.

Interactive pattern classification was first proposed more than forty years ago: "Starting with the era of learning machines, reasons are presented for the current emergence of graphics-oriented interactive pattern analysis and classification systems (IPACS) as a general approach to practical pattern-recognition problems." [10]. It was, however, seldom coupled with (re-)trainable classifiers. Most of the early work was directed at exploration of multidimensional feature spaces through the then available graphics [11].

Active learning, where samples to be labeled are selected automatically, has goals similar to CAVIAR's [12]. A comparison between these approaches, based on the human time required to produce an accurately labeled dataset, would be timely.

Only a few papers, even among those that address error-sensitive applications, discuss what is to be done with residual errors [13]. An application where complete verification is indispensable is financial and medical form entry [14].

3. A model for determining the best splits

We model a multistage labeling process where the errors at each stage *i* are corrected by a human operator and the samples are added to the previous training set to retrain the classifier. The classifier trained on N_{i-1} samples makes M_i errors when it classifies the next N_i samples ($i = 1,2,\ldots, n$). These errors are corrected and the rest of the labels are approved.

The average duration of inspecting and confirming a correct label is taken as the unit of time or cost: each confirmation has unit cost. Either entering a new label or correcting a wrong label is assumed to take *r* units of time and have cost *r*. The number of errors M_i at the i^{th} stage is $N_i\, f(N_{train}\, ;\, \beta)$, where β is a scalar or vector parameter of the function that specifies the dependence of the error rate on the training set size. The time required to confirm all the correct labels at the i^{th} stage is $(N_i - M_i)$, and the time required to correct the errors is $r \times M_i$.

All N_0 objects of the initial training set must be labeled, therefore $M_0 = N_0$. The operator time at stage *i* is T_i. More formally:

$$M_0 = N_0,\quad n > 0,\ f(0;\sigma) = 1,\ r \geq 1,$$
$$\text{and}\quad N_{total} = \textstyle\sum_{k=0}^{n} N_k \,.$$

$$M_i = N_i f\left(\sum_{k=1}^{i} N_{k-1}\,; \beta\right)$$

$$T_i = (N_i - M_i) + rM_i = N_i + (r-1)M_i$$
$$= N_i\left(1 + (r-1) f\left(\sum_{k=1}^{i} N_{k-1}\,; \beta\right)\right)$$

$$T_{total} = \sum_{h=0}^{n} T_h$$
$$= rN_0 + \sum_{h=1}^{n} N_h \left(1 + (r-1) f\left(\sum_{k=1}^{h} N_{k-1}; \beta\right)\right)$$

For $n = 2$ it is easy to show that the derivative of (T_0+T_1) with respect to N_0 is proportional to $(r\text{-}1)$. Therefore the optimum split is independent of *r*. This holds also for $n > 2$. The total cost and the error function $e^{-\beta N_{train}}$, with n = 1, $N_{total} = 1000$, $r = 3$, and $\beta = 0.0046$, are graphed in Figure 2 as a function of N_0. The optimal N_0 is *311*, so N_1 is *689*. Splitting at $N_0 = 500$ would increase interaction time by 8%.

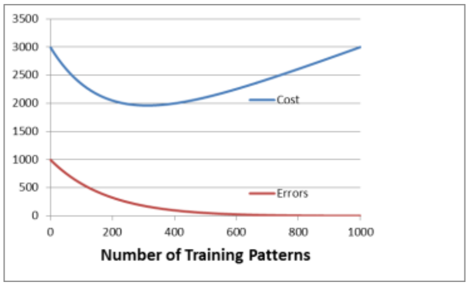

Fig. 2. Cost (time) and number of errors as a function of the size of the training set

We resorted to exhaustive search to find the optimum number of samples before switching from automatic processing to interaction for $n = 3$. We set the parameter β to reach an error rate of either $E_{1000} = 1\%$ or $E_{1000} = 5\%$ after training on *1000* patterns.

Published experiments indicate that the effect of the size of the training set on the error rate varies considerably depending on the data, features, and classifier. We have therefore calculated the splits for super-linear, linear, and supra-linear initial fall-off of the error rate with the size of the training set. These functions, graphed in Figure 3, are:

$$f\,(N_{train};\,\beta) \;=\; e^{-\beta N_{train}};$$
$$g\,(N_{train};\,\beta) = 1 - \beta\, N_{train}$$
$$h\,(N_{train};\,\beta) = 1 - \beta\,(N_{train})^2\,.$$

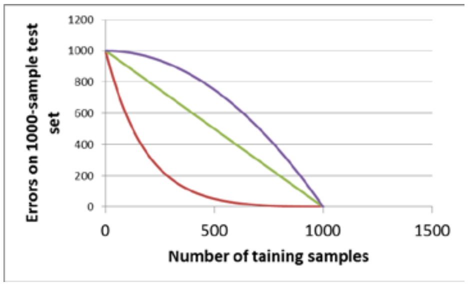

Fig. 3. Error functions *f, g*, and *h* as a function of the size of the training set

The values of N_0, N_1, and N_2 required to minimize T_{total} are shown for several situations in Table II. $N_{total} = 1000$ in every simulation except the last one, where $N_{total} = 5000$. The human time saved by using the classifier is $S = 1 - T_{total} / T_{max} = 1 - T_{total} / r\, N_{total}$.

If correction and verification cost the same ($r = 1$), any choice would give the same final cost $r\ N_{total}$. The slower the error rate decreases initially, the more samples should be used in the earlier stages. Poor initial generalization leads to higher final cost and less savings. A linear error function splits the data uniformly. The gain over direct entry increases with the ratio of corrections to verification and with the total number of samples to be classified. With error function *f*, increasing the total sample size from 1000 to 5000 almost doubles the time saved.

Table II. Optimal splits and final costs

error fn	***E1000***	***r***	***N_0***	***N_1***	***N_2***	***S(%)***
h(•)	0.01	1	0	0	1000	**0**
f(•)	0.01	3	177	274	549	**73**
g(•)	0.01	3	333	333	334	**25**
h(•)	0.01	3	645	237	118	**11**
f(•)	0.05	3	214	300	486	**54**
g(•)	0.05	3	333	333	334	**23**
h(•)	0.01	5	645	237	118	**14**
f(•)	0.05	3	418	836	3746	**131**

4. Discussion

The above analysis provides guidance to optimal sizing of data sets when human labeling alternates with automated classifier training and labeling. Practical application requires an approximation of the error function. Fortunately, only an approximation of the early, high-error segment is needed, and the approximation can be improved from the results on the early stages. The relative cost of corrections vs. confirmations does not affect the experimental protocol, only the overall gain.

The gain is a slowly rising function of the number of stages, which in practice is limited by the logistics of retraining the classifier and scheduling interaction. Ideally the classifier would be retrained after every verified or corrected label.

Some aspects of the proposed protocol that deserve further consideration are:

4.1 Domain of application

The current analysis is limited to classifiers that treat each pattern as an independent entity. It excludes classifiers that use language, scene, or style context because such contexts may extend beyond batch boundaries.

In many applications, however, only-short range context is exploited. If this range is much smaller than the number of samples in the split sets, then the model may still apply.

4.2 Fundamental independence assumption

We have implicitly assumed that the classification process is stationary and that successive decisions are statistically independent. The best way to assure that these assumptions apply to any particular set of data is to reorder the entire data set with a pseudo-random permutation. That guarantees that each training set is representative of the corresponding test set, and that successive classifier decisions are statistically independent. Although on any particular sequence the error rate may not decrease monotonically, randomization ensures that it does so on average.

In contrast to experiments designed to demonstrate generalization, the grain of randomization should be as small as possible in order to minimize errors. In document image analysis, for example, we would randomly split each document into individual glyphs or words. In table processing or writer recognition, each source should be split into sequences of samples.

4.3 Statistical fluctuation of the error rate

The optimality of the splits applies of course only to statistical expectations rather than individual runs. Fortunately the optimal splits seldom push the classifier to its lowest error rate on the available sample size. For instance, the error rate at the optimal N_0 in Fig. 2 is 24%, in contrast to 1% at $N_0 = 1000$. The statistical fluctuation of the higher error rates resulting from medium-sized training sets tends to be much less than those generally reported for cross-validation using the largest possible sets of training samples.

4.4 Cost of simulation

For a three-stage system for 1000 samples, the simulation runs on a garden-variety laptop in less than a tenth of a second. It is, however, proportional to $(N_{total})^{n-1}$, where n is the number of stages, because it searches over the cross product of every stage except the last (the splits must add up to N_{total}).

There are several means of reducing run time. If the error function is monotonically decreasing, the total time has only a single minimum. Therefore any gradient-descent method will find it quickly.

Furthermore, the only critical input to the simulation is the error function. Instead of searching for the exact number of samples in each split, if we have a million samples we can scale the error function to 1000 or 10,000 samples and search for splits accurate to the nearest 1000 or nearest 100 samples. Of course interaction on such large data sets raises other problems as well.

4.5 Selective inspection

It is appealing to consider inspecting and correcting or verifying only the labels of patterns flagged by the classifier (as in [1]). This procedure, however, fails to guarantee a final error rate of zero (i.e., labels considered correct by a human operator.).

If each stage is not completely verified and only rejects or patterns flagged “critical” by the classifier are labeled by the operator, then some cost must be assigned to undetected errors relative to rejects. The simulation can be readily modified to accommodate any available error/reject or ROC curve. This would be appropriate in applications where the cost of exhaustive verification is prohibitive.

Acknowledgment

Daniel Lopresti receives support from a DARPA IPTO grant administered by Raytheon BBN Technologies.

References

[1] R.N. Ascher, G. Koppelman, M.J. Miller, and G. Nagy, "An Interactive System for Reading Unformatted Printed Text," IEEE Transactions on Computers, vol. 20, #12, pp. 1527-1543, December 1971.

[2] Y. Xu and G. Nagy, "Prototype Extraction and Adaptive OCR," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, #12, pp. 1280-1296, December 1999.

[3] J. Zou and G. Nagy, "Evaluation of model-based interactive pattern recognition," Proceedings of International Conference on Pattern Recognition XVII, vol.II, pp. 311-314, Cambridge, UK, August 2004.

[4] A. Evans, J. Sikorski, P. Thomas, S-H Cha, C. Tappert, G. Zou, A. Gattani, and G. Nagy, “Computer Assisted Visual Interactive Recognition (CAVIAR) Technology,” 2005 IEEE International Conference on Electro-Information Technology, Lincoln, NE, May 2005.

[5] G. Nagy, X. Zhang, CalliGUI: “Interactive Labeling of Calligraphic Character Images,” Procs. ICDAR 11, Beijing, September 2011.

[6] G. Nagy, M. Tamhankar, “VeriClick, an efficient tool for table format verification,” Procs. SPIE/EIT/DRR, San Francisco, Jan. 2012.

[7] J. Zou and G. Nagy, “Visible models for interactive pattern recognition,” Pattern Recognition Letters Vol. 28, pp 2335-2342, 2007.

[8] D. Lopresti, G. Nagy, E. Barney Smith, “,” Proceedings of the International Workshop on Document Analysis Systems (DAS20), pp. 105-112, Boston, June 2010.

[9] Y. Zhu, T. Shen, D. Lopresti, X. “Huang: Interactive Polygons in Region-Based Deformable Contours for Medical Images,” Int. Conf. Biomedical Imaging,(ISBI_ Boston, 37-40, 2009.

10 L. Kanal, “Interactive pattern analysis and classification systems: A survey and commentary,” Proceedings of the IEEE, 60, 10, 1200-1215, Oct. 1972.

11 J.W. Sammon, “Interactive pattern analysis and classification,” IEEE Trans. Computers C-19, 7, 594-616, July 1970.

[12] A. Cohn, Z. Ghahramani, M.I. Jordan, Active Learning with Statistical Models, J. Artificial Intelligence Research, Vol. 4, 129-145, 1996.

[13] K. Taghva, E. Stofsky, “OCRSpell: an interactive spelling correction system for OCR errors in text,” IJDAR 3,3, 125-`137, 2001.

[14] B. Klein, A. Dengel,” Problem-adaptable document analysis and understanding for high-volume applications,” IJDAR 6, 3, 167-180, 2003.