Discovering Informative Patterns and Data Cleaning

I. Guyon*, N. Matić, and V. Vapnik

AT&T Bell Laboratories, Holmdel, NJ 07733
* AT&T, 50 Fremont street, 40th floor,
San Francisco, CA 94105,
isabelle@neural.att.com

Abstract

We present a method for discovering informative patterns from data. With this method, large databases can be reduced to only a few representative data entries. Our framework encompasses also methods for cleaning databases containing corrupted data. Both on-line and off-line algorithms are proposed and experimentally checked on databases of handwritten images. The generality of the framework makes it an attractive candidate for new applications in knowledge discovery.

Keywords: knowledge discovery, machine learning, informative patterns, data cleaning, information gain.

1 INTRODUCTION

Databases often contain redundant data. It would be convenient if large databases could be replaced by only a subset of informative patterns. A difficult, yet important problem, is to define what informative patterns are. We use the learning theoretic definition [11, 6, 8]: given a model trained on a sequence of patterns, a new pattern is informative if it is difficult to predict by a model trained on previously seen data. With that definition, we derive on-line and batch algorithms for discovering informative patterns. The techniques were developed for classification problems, but are also applicable to regression and density estimation problems.

Informative patterns are often intermixed with other "bad" outlyers which correspond to errors introduced non-intentionally in the database. For databases containing errors, our algorithms can be used to do computer-aided data cleaning, with or without supervision. We review several results of experiments in handwriting recognition [1, 5, 9] which demonstrate the usefulness of our data cleaning techniques.

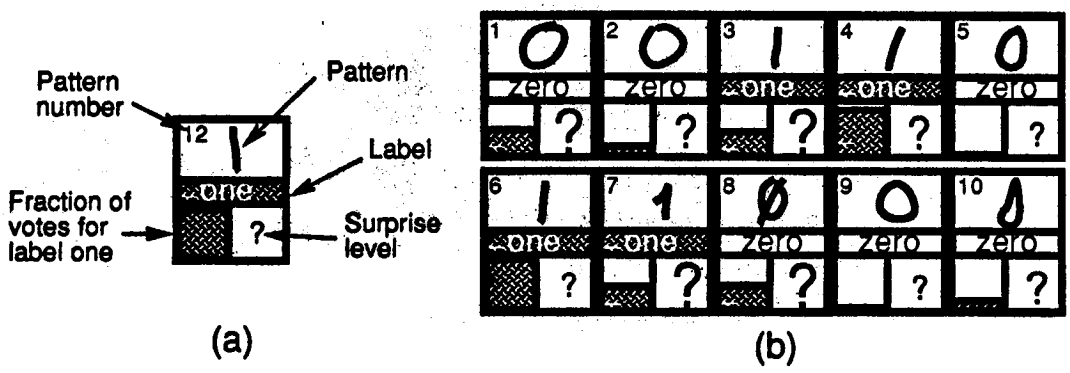

Figure 1: Small example database containing "zeros" and "ones". (a) A data entry. (b) A sequence of data entries during a training session showing the variation of the surprise level. The patterns which are most surprising are most informative.

2 DISCOVERING INFORMATIVE PATTERNS

In this section, we assume that the data is perfectly clean. First, we give an intuition of what informative patterns ought to be. Then, we show that this intuition coincides with the information theoretic definition. Finally, we derive algorithms to discover informative patterns.

2.1 Informative Patterns are most Surprising

In figure 1, we constructed a small example database containing only handwritten zeros and ones. Most patterns of a given category look similar. A typical zero is a circle and a typical one is a vertical bar. However, there exist other shapes of zeros and ones. If we wanted to keep only a few data representatives, we would probably keep at least one example of each basic shape.

To choose the best data representatives, we run an imaginary experiment. Imagine that we found 100 people who did not know at all what the shape of a zero and that of a one are. We teach those people to recognize zeros and ones by letting them examine the patterns of our database in sequence. Every time we show them a new image, we first hide the label and let them make a guess.

We show in figure 1 the average value of the guesses for a particular sequence of data. Since we assumed that our subjects had never seen a zero nor a one before the experiment, about 50% guessed "zero" and 50% guessed "one" when they were shown the first pattern. But, for the second example of the same shape, the majority made the correct guess. As learning goes on, familiar shapes are guessed more and more accurately and the percentage of wrong guesses raises only occasionally when a new shape appears.

We represent with the size of a question mark the average amount of surprise that was generated among our subjects when the true label was uncovered to them. People who guessed the correct label where not surprised while people who made the wrong guess were surprised. We see on figure 1 that a large average surprise level coincides

with the apparition of a new shape. Therefore, the level of surprise is a good indication of how informative a pattern is.

More formally, the level of surprise varies in the opposite direction as the probability of guessing the correct label $P_k(\hat{y}_k = y_k) = P(\hat{y}_k = y_k | x_k; (x_0, y_0), (x_1, y_1), ...(x_{k-1}, y_{k-1}))$[1]. This probability of making the correct guess is precisely what is involved in Shannon's information gain:

$$I(k) = -\log P_k(\hat{y}_k = y_k) = -y_k \log P_k(\hat{y}_k = 1) - (1 - y_k) \log(1 - P_k(\hat{y}_k = 1)) \quad (1)$$

where x_k is an image, $y_k \in \{0, 1\}$ is its associated label, $k - 1$ is the number of data entries seen thus far and $\hat{y}_k$ is the label predicted for pattern x_k. The log dependency ensures additivity of information quantities. In the information theoretic sense, the data entries that are most informative are those that are most surprising.

2.2 Machine Learning Techniques to Estimate the Information Gain

It is somewhat unrealistic to hire 100 ignorant people to estimate the information gain. Let us now replace people with machines.

In the Machine Learning framework, patterns drawn from a database are presented to the learning machine which makes predictions. The prediction error is evaluated and used to improve the accuracy of further predictions by adjusting the learning machine parameters.

Assume first that we trained 100 different learning machines (substituting our 100 people), each one predicting its own value of $\hat{y}_k$. This is referred to as a "Bayesian" approach [8]. Formula 1 can be readily used to determine the information gain. Although this is a perfectly valid method, we propose here a more economical one: we train a single learning machine to give an estimate $\hat{P}_k(y_k = 1)$ of the probability that the correct label is "one" (figure 2). Our prediction $\hat{y}_k$ will me the most likely category according to $\hat{P}_k(y_k = 1)$. In formula 1, we substitute $\hat{P}_k(y_k = 1)$ to $P_k(\hat{y}_k = 1)$.

Many Machine Learning techniques can be applied to discover informative patterns. For example, the learning machine can be a simple K-nearest-neighbor classifier [4]. All patterns presented to the classifier are stored. $\hat{P}_k(y_k = 1)$ is given by the fraction of the K training patterns that are nearest to x_k which have label "one".

Another example is a neural network trained with a "cross-entropy" cost function for which the information gain is the cost function itself. The mean square error cost function $(y_k - \hat{P}_k(y_k = 1))^2$ provides an *information criterion* which ranks patterns in the same order as the cross-entropy cost function and can therefore be used as well.

In the following, we use the size of the symbol "question mark", in the figures, and the notation I(k), in the text, to represent the *information criteria* used to measure the "surprise level".

[1] To be perfectly correct, the probability should be also conditioned on the model class.

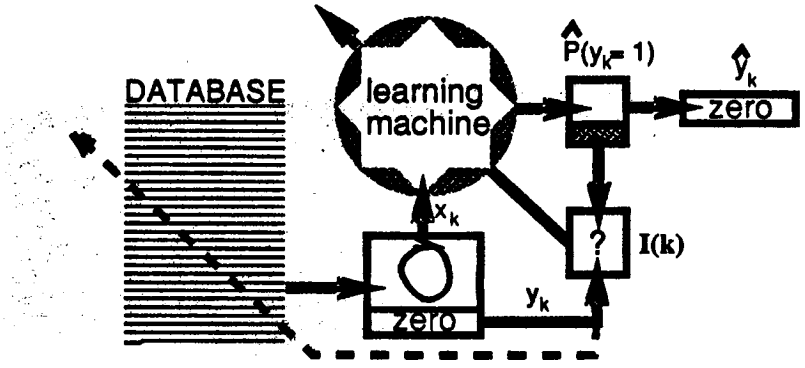

Figure 2: A learning machine used to predict the information gain (or surprise level). In the classical machine learning framework, the goal is to train the learning machine, either to provide a model of the data or to make predictions. In our framework the goal is to discover the informative patterns of the database (dashed line).

2.3 On-line Algorithms and Batch Algorithms

In an on-line algorithm, patterns are presented in sequence and the learning machine adjusts its parameters at each presentation of a new pattern. This is a situation similar to that of our example of section 2.1. In that case, we will say that a pattern is informative if the information gain exceeds a pre-defined threshold. The disadvantage of this method is that informative patterns depend on the sequence in which patterns are presented. However, there may be practical situations where the data is only available on-line.

In a batch algorithm, conversely, all data entries are available at once and the information gain is independent on pattern ordering. This implies that, if there are p data entries in the database, we need to train p machines, each one on all the data but one pattern, and then try to predict that last pattern. In practice, this is feasible only if training is unexpensive, as for the K-nearest-neighbor algorithm.

For other batch algorithms, we rather train the learning machine only once on all the data. We then approximate the information gain of each pattern with an estimate of how much the cumulative information gain would decrease if we removed that pattern from the training set.

For batch algorithms all the patterns of the database are uniquely ranked according to their information gain. The m most informative patterns can be selected to represent the entire database.

2.4 Minimax Algorithms

Minimax algorithms are batch algorithms that are particularly well suited to discover informative patterns. Most algorithms train the learning machine to minimize the average loss (e.g. the mean-square-error). Minimax algorithms minimize the maximum loss:

$$\min_w \max_k J(k) \tag{2}$$

where w represents the parameters of the learning machine and k runs over all data entries. Minimax algorithms are extreme cases of some "active learning" methods which emphasize the patterns with large information gain [8]. The solution of a minimax algorithm is a function of only a small subset of the training patterns, precisely called "informative patterns" [12]. These are the patterns that have maximum loss.

In reference [1], we propose and study a minimax algorithm for classification problems: the Optimum Margin Classifier. The algorithm maximize the minimum distance of the training patterns to the decision boundary. It is shown that the solution w^* is a linear combination of basis functions of the informative patterns:

$$w^* = \sum_{k=1}^{p} (2y_k - 1)\alpha_k \varphi(x_k), \quad \alpha_k \geq 0, \tag{3}$$

where $y_k \in \{0, 1\}$ indicates the class membership and the α_k coefficients are all zeros, except for the informative patterns. We use the value of the cost function at the solution as cumulative information criterion:

$$I = \sum_{k=1}^{p} \alpha_k, \tag{4}$$

from which we derive and estimate of the information loss incurred by removing the informative pattern k:

$$I(k) = \alpha_k. \tag{5}$$

One important question is: what is the rate of growth of the number of informative patterns with the size of the database. The results of experiments carried out on a database of handwritten digits using the Optimal Margin Classifier suggest a logarithmic growth, in the regime when the number of patterns is small compared to the total number of distinct patterns (figure 3).

Other minimax algorithms have been proposed for classification and regression [3, 2].

3 DATA CLEANING

In this section we tackle the problem of real world databases which may contain corrupted data entries. We propose data cleaning algorithms and analyze experimental results.

3.1 Garbage Patterns are also Surprising

The information theoretic definition of informative pattern does not always coincide with the common sense definition. In figure 4, we show examples of patterns drawn from our database of "zeros" and "ones", which have a large information gain. We see two kinds of patterns:

- Patterns that are actually informative: Atypical shapes or ambiguous shapes.
- Garbage patterns: Meaningless or mislabeled patterns.

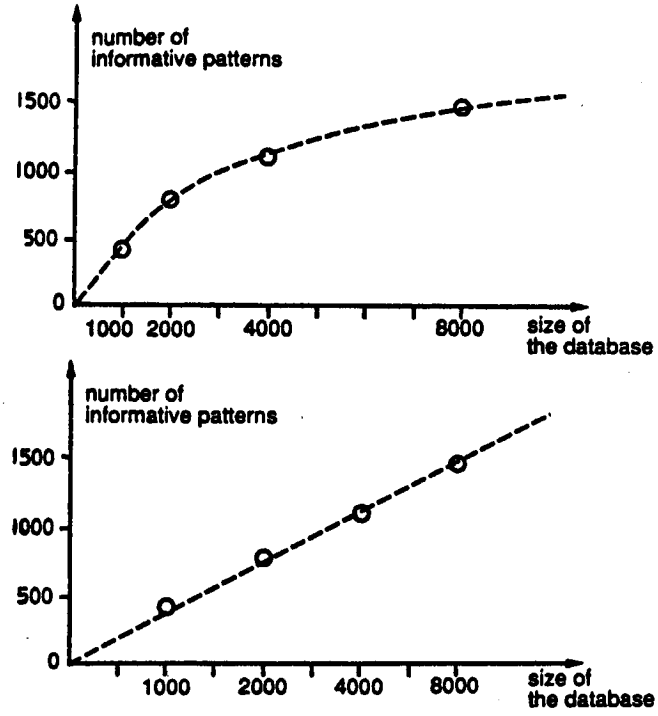

Figure 3: Variation of the number of informative patterns as a function of the number of patterns in the database. Experiments were carried out on a database of handwritten digits, encoded with 16 real-valued global features derived from the pen trajectory information. A polynomial classifier of degree 2 was trained with a minimax algorithm (Optimal Margin Classifier). The informative patterns are the patterns for which $I(x_k) = \alpha_k \neq 0$.

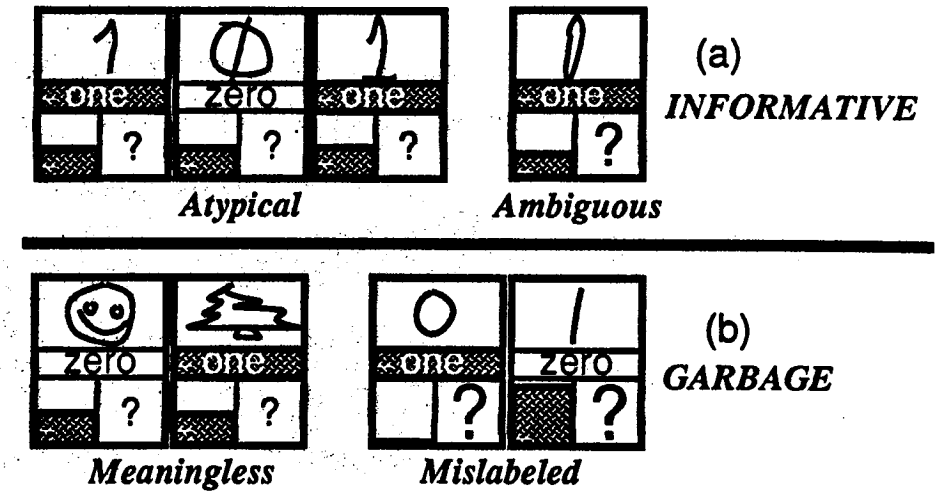

Figure 4: (a) Informative patterns versus (b) garbage patterns. Informative patterns are intermixed with garbage patterns which also generate a lot of surprise (i.e. have a large information gain).

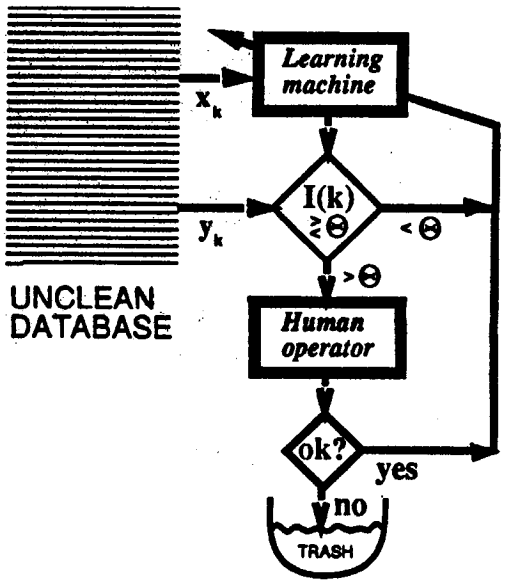

Figure 5: Flow diagram of on-line cleaning. In the process of learning, patterns which information gain exceeds threshold θ are examined by a human operator. Good patterns are kept in the database and sent to the recognizer for adaptation. Bad patterns are removed from the database.

Truly informative patterns should be kept in the database while garbage patterns should be eliminated.

Purely automatic cleaning could be performed by eliminating systematically all patterns with suspiciously large information gain. However, this is dangerous since valuable informative patterns may also be eliminated. Purely manual cleaning, by examining all patterns in the database, is tedious and impractical for very large databases. We propose a computer-aided cleaning method where a human operator must check only those patterns that have largest information gain and are therefore most suspicious.

3.2 On-line Algorithms and Batch Algorithms

In figure 5 we present our on-line version of data cleaning. It combines cleaning and training in one single session. The learning machine is initially trained with a few clean examples. At step k of the cleaning process, a new pattern x_k is presented to the learning machine. The prediction of the learning machine and the desired value y_k are used to compute the information criterion $I(k)$. If $I(k)$ is below a given threshold θ, the pattern is directly sent to the learning machine for adaptation. Otherwise, the pattern is sent to the human operator for checking. Depending on the decision of the operator, the pattern is either trashed or sent to the learning machine for adaptation.

When all the data has been processed, both training and cleaning are completed, since the learning machine was trained only on clean data. This is an advantage if further use is made of the learning machine. But on-line cleaning has several disadvantages:

- One has to find find an appropriate threshold θ.

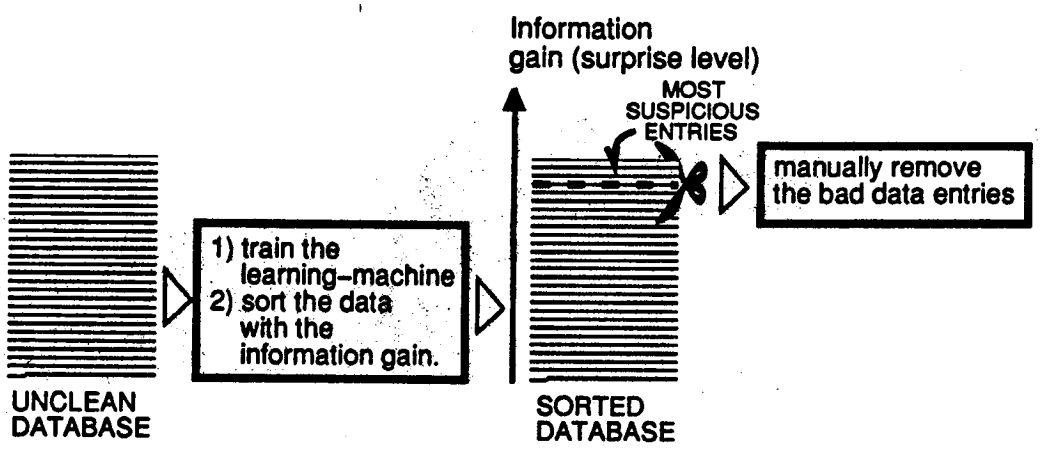

Figure 6: Block diagram of batch cleaning. The learning machine is trained on unclean data. The information gain is evaluated for each pattern by estimating how much information would be lost if that pattern would be removed. The database is sorted sorted according to the information gain. Only the top ranked patterns are examined for cleaning by the human operator. After cleaning, the learning machine needs to be retrained to obtain a good model and/or make good predictions.

- Training the learning machine may be slower than checking the patterns which may result in wasting the time of the operator.
- The method depends on the order of presentation of the patterns; it is not possible to revert a decision on whether a pattern is good or bad.

When possible, batch methods are preferred for cleaning (figure 6). Training is performed on all patterns, including garbage patterns. The information gain of the patterns is computed as explained in sections 2.3 and 2.4. The data entries are then sorted in decreasing order of information gain. The patterns are examined by the operator, starting from the top (most suspicious), and until the number of consecutive "good" patterns exceeds a given threshold. If the database contains correlated errors, it may be necessary to iterate this procedure several times to remove all "bad" patterns.

The combination of batch cleaning and minimax algorithms is particularly attractive. In that case, only informative patterns (with non-zero information gain) need to be examined. In figure 7 we show the first few informative patterns obtained with a minimax classifier (the Optimum Margin Classifier [1]). The classifier was trained to discriminate between the digit "two" and all the other digits. The patterns with largest α_k (our estimate of the information gain from equation 5) is a garbage pattern.

3.3 Point of Optimal Cleaning

We may wonder how reliable these cleaning techniques are: Did we examine all the patterns that should be candidates for cleaning? Among the patterns that were candidates for cleaning, did we remove too many or too few patterns?

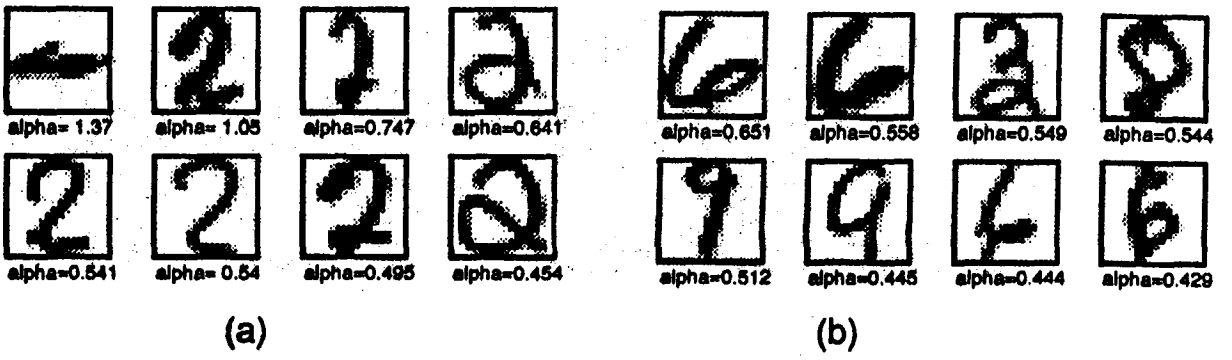

Figure 7: The informative patterns obtained by a minimax algorithm (the Optimum Margin Algorithm), for the separation of handwritten digit "two" against all other digit categories [1]. Patterns are represented by a 16x16 grey-level pixel-map. The informative patterns are shown in order of decreasing information gain.(a) Informative patterns for class 2: a garbage pattern comes first. (b) Informative patterns for all other classes: several ambiguous or atypical shapes are among the patterns with largest information gain.

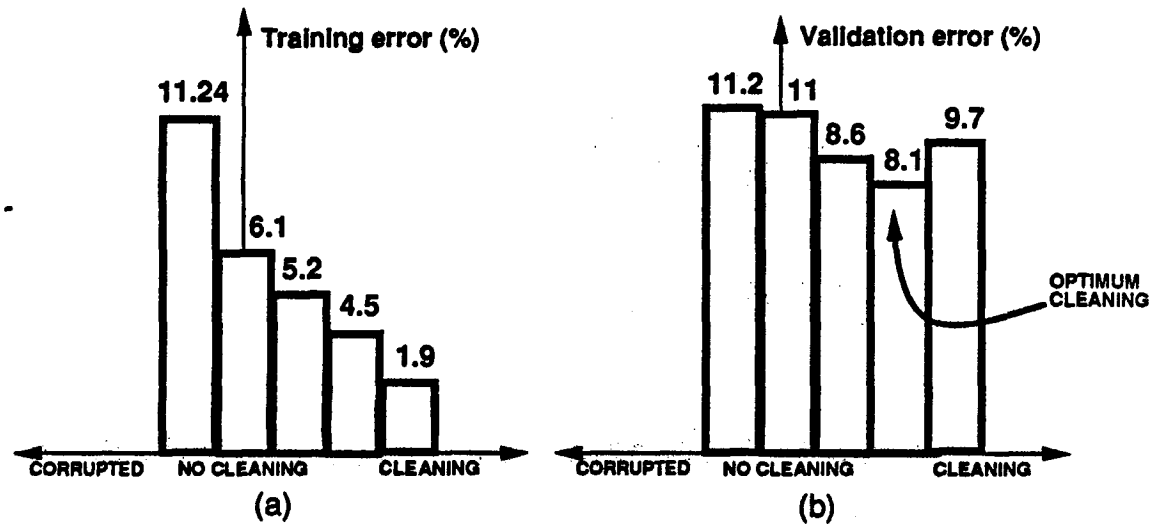

Figure 8: Data cleaning results showing a point of optimal cleaning [9]. A neural network was trained with a mean-square-error cost function to recognize handwritten lowercase letters. The representation is 630 local features of the pen trajectory. A training data set of 9513 characters and a validation data set of 2000 characters from disjoint set of writers was used for cleaning. Starting with the "uncleaned" database, several levels of cleaning were applied by the human operator. Each stage was more strict, i.e. he lowered the tolerance for marginal-quality characters. To test the power of the cleaning technique, we also corrupted the initial, uncleaned database, by artificially mislabeling 5% of the characters in the training set (left most column).

We can again use our learning machine to provide us with an answer. In the experiments of figure 8, we varied the amount of cleaning. The data was split into a training set and a validation set. The learning machine was trained on data from the training set with various amounts of cleaning. It was tested with the unclean data of the validation set. We observe that, as more patterns are removed through the cleaning process, the error rate on the training set decreases. This is understandable since the learning task becomes easier and easier to the learning machine. The validation error however goes through a minimum. This is because we remove first only really bad patterns, then we start removing valuable informative patterns that we mistook for garbage patterns. The minimum of the validation error is the point of optimal cleaning. If our validation error does not go through a minimum, more patterns should be examined and considered for cleaning.

It is not always possible nor desirable to split the database into a training and a validation set. Another way of obtaining the point of optimum cleaning is to use the predictions of the Vapnik-Chervonenkis (VC) theory [12]. According to the VC-theory, the point of optimum cleaning is the minimum of the so-called "guaranteed risk", which is a bound on the validation set frequency of errors:

$$G = E_{train}(m) + \beta \frac{d(ln\frac{2(p-m)}{d} + 1) + m(ln\frac{p}{m} + 1)}{p - m} \qquad (6)$$

where β is a constant which was experimentally determined with the method described in reference [7] ($\beta = 0.5$), $E_{train}(m)$ is the frequency training errors when m suspicious patterns have been removed from the training set, p is the number of training patterns before cleaning (p=9513) and d is the VC-dimension.

These predictions can be made only if the capacity (or VC-dimension) of the learning machine is known and if the training algorithm does not learn the training set with zero error. In our experiments on data cleaning with neural networks (figure 9) [9] we obtained good agreement between the prediction of the VC-theory and that of the validation set, even with a very rough estimate of the VC-dimension. In fact, we checked the robustness of the VC-prediction of optimum cleaning with respect to a change in the estimate of the VC-dimension. We found no significant change in the position of the optimum in the range $300 < d < 5000$.

4 CONCLUSIONS

We presented a computer-aided method for detecting informative patterns and cleaning data. We used this method on various databases of handwritten characters. The results are summarized in table 1. It is important to stress that the number of informative patterns varies sub-linearly with the number of data entries and that therefore the cleaning time varies sub-linearly with the number of data entries. The point of optimum cleaning can be predicted by the VC-theory which does not require splitting the data between a training set and a validation set. Our cleaning method clearly improves the quality of the data, as measured by the improvement in classification performance on a test set independent from the training set. Our framework is general and applies to other problems than classification problems (regression or density estimation) which makes it attractive for new applications in knowledge discovery.

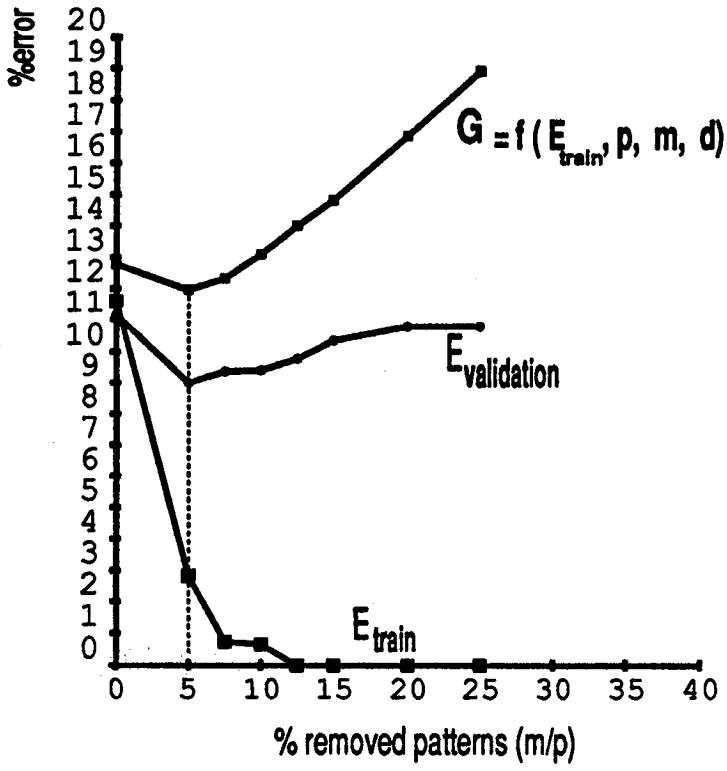

Figure 9: Detection of the point of optimal cleaning with the Vapnik-Chervonenkis prediction. A neural network was trained to minimize the maximum squared error. Training and validation sets are the same as in figure 8. The VC-dimension of the neural network was estimated to d=2000 (approximately one third of the number of free parameters). G has been rescaled to fit on the figure.

References	Database	Db. size	Num. of info. patt.	Cleaning time (h)	% test error unclean data	% test error clean data
Boser-92[1]	OCR digits	7,300	690	1	15	10.5
Matić-93[10]	on-line lower	16,000	3,130	1	11	6.5
Guyon-92	on-line ASCII	100,000	10,983	5	30	6

Table 1: Results obtained on various databases of handwritten patterns. The cleaning time is the time spent by the operator browsing through the informative patterns (not including the learning machine training time). The test errors were computed on separated test sets of more than 2000 examples from a disjoint set of writers. Without cleaning, the last database was completely worthless.

Acknowledgements

Discussions with colleagues AT&T Bell Laboratories and at UC Berkeley are gratefully acknowledged. Special thanks to Bernhard Boser who contributed many ideas.

References

[1] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, July 1992. ACM.

[2] C. Cortes and V. Vapnik. Soft margin classifier. Technical report, AT&T Bell Labs, Holmdel, NJ, December 1992.

[3] V. F. Demyanov and V. N. Malozemov. *Introduction to Minimax.* Dover, New York, 1972.

[4] R.O. Duda and P.E. Hart. *Pattern Classification And Scene Analysis.* Wiley and Son, 1973.

[5] I. Guyon, B. Boser, and V. Vapnik. Automatic capacity tuning of very large VC dimension classifiers. In *Neural Information Processing Systems conference*, pages 147–155, Denver, December 1992.

[6] D. Haussler, M. Kearns, and R. Shapire. Bounds on the sample complexity of bayesian learning using information theory and the VC dimension. In *Computational Learning Theory workshop, ACM, 1991. Also Machine Learning*, volume 14 (1), pages 83–113, January 1994.

[7] E. Levin, Y. Le Cun, and V. Vapnik. Measuring the capacity of a learning machine. Technical Report 11359-920728-20TM, AT&T Bell Labs, Holmdel, NJ, 1992.

[8] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4 (4):590–604, July 1992.

[9] N. Matić, I. Guyon, L. Bottou, J. Denker, and V. Vapnik. Computer aided cleaning of large databases for character recognition. In *11th International Conference on Pattern Recognition*, volume II, pages 330–333, Amsterdam, August 1992. IAPR/IEEE.

[10] N. Matić, I. Guyon, J. Denker, and V. Vapnik. Writer adaptation for on-line handwritten character recognition. In *Second International Conference on Pattern Recognition and Document Analysis*, pages 187–191, Tsukuba, Japan, October 1993. IAPR/IEEE.

[11] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific, Singapore, 1989.

[12] V.N. Vapnik and A.Ya. Chervonenkis. *The theory of pattern recognition.* Nauka, Moscow, 1974.