Open data collection for training intelligent software in the Open Mind Initiative

David G. Stork
Ricoh Silicon Valley
2882 Sand Hill Road Suite 115
Menlo Park, CA 94025-7022 USA
stork@OpenMind.org*

Abstract

The Open Mind Initiative is an internet based collaborative framework for constructing intelligent systems. Open Mind extends traditional open source development methods by allowing non-expert "netizens" to contribute informal data by means of interactive queries presented on web browsers. Since this data is used to train classifiers or guide automatic inferencing systems, it is important to accept data of high quality and consistency and reject data of low quality. We identify a number of forms of low-quality data from unreliable contributors to a specific Open Mind demonstration program, *Animals*, as well as in a broad class of pattern recognition projects. We tested several software modules that automatically or semi-automatically reduce the effects of poor data in *Animals*. We also discuss a number of techniques for ensuring data quality that can be tailored to generic pattern recognition and artificial intelligence projects. These techniques possess parameters that can be set to give the optimal balance between data *quality* and data *quantity* to give the fastest improvement of the system.

1 Introduction

The vast majority of intelligent systems are built using models (expert systems, case based reasoners, Bayes belief networks, neural networks, etc.) as well as domain specific data. For some cases, this data must be provided by experts, as in automated medical diagnosis systems [12]. In many other cases, such data can be collected from non-experts, for instance some forms of speech data. The more high-quality data that is used, the better the resulting system. For example, handwriting recognition systems based on even simple models such as the nearest-neighbor algorithm or decision trees can perform with high accuracies if they are trained with very large corpora [6, 2]. Likewise, the accuracy of acoustic speech recognizers improves with increasing size of the training set [7].

The World Wide Web represents an enormous passive source of data; it has been exploited in many data mining projects [5]. The central hypothesis underlying the Open Mind Initiative is that for the development of some classes of intelligent software, data can be contributed *actively* and *willingly* by a large number of non-expert netizens over the internet. Thus Open Mind seeks to leverage the large

* *Proceedings of the Engineering Intelligent Systems Symposium* (EIS2000), Paisley, Scotland, June 2000.

number of netizens and reduced cost of data collection afforded by the internet. Our concern in this paper is a central problem in Open Mind arising from the variation in netizen reliability: how to ensure that the unrejected data has an acceptably small number of errors and represents a consensus of the participating netizen population.

2 Open Mind Initiative

In broad overview, the Open Mind Initiative is an internet based collaborative framework for developing intelligent software such as speech and handwriting recognizers, common sense reasoning systems, smart spam filters, and so on [14, 15, 18]. The Initiative relies on three general forms of contribution: 1) *domain experts* provide fundamental algorithms such as learning algorithms for character recognition, 2) *infrastructure/tool developers* provide software infrastructure for capturing raw data and rewarding netizens, and 3) non-expert *netizens* contribute raw data over the internet, for instance character labels.[1] The incentives for netizens to contribute include public acknowledgement on the Open Mind website, altruism and inherent interest in artificial intelligence and particular project domains, pleasure from playing games (serving as interfaces), lotteries, gifts and coupons donated by corporations, and more [14]. Experts can propose changes to the source code which, along with the data, are made available through open source licenses.

The Initiative arose from a deep appreciation of the following facts and recent trends. 1) The increasing acceptance of open source development methods and resulting software such as *Linux*, *Mozilla*, *emacs* and *Apache*. 2) The refinement of highly developed techniques in pattern recognition, machine learning, grammatical inference, data mining and closely related core disciplines [4]. 3) The realization that many problems in pattern recognition and intelligent systems require very large data sets and that these can be provided by non-experts. 4) The increase in collaboration over the internet and the improvement of tools facilitating such collaboration among experts and non-experts alike. 5) The growth in the participation of non-experts (e.g., non-programmers) in group projects over the internet such as the Search for Extraterrestrial Intelligence[2] and Great Internet Mersenne Prime Search.[3] In other collaborative projects, netizens contribute non-expert or "informal" information, as in the Newhoo! open web directory (Table 1).[4]

Open Source	**Open Mind**
no netizens	netizens crucial
expert knowledge	informal knowledge
no machine learning	machine learning
navigation of data (e.g., Newhoo!)	single classifier/AI system (e.g., OCR)
most work is directly on the final software	most work is on data collection/learning
$\approx 10^5$ *Linux* hackers	$\approx 10^8$ netizens
many functs. contributed (e.g., drivers)	one functional goal (e.g., OCR accuracy)
software licenses	S/W & data licenses

Table 1: Comparison of traditional open source and Open Mind approaches.

In contradistinction to traditional data mining techniques [5], Open Mind affords interactive learning, such as learning with queries in which the *most informative* data is requested from contributors. Such learning is generally faster than non-interactive learning based on randomly sampled data [1].

[1] www.OpenMind.org

[2] setiathome.ssl.berkeley.edu

[3] www.mersenne.org

[4] www.dmoz.org

Consider the collecting of training data for a handwriting recognizer by means of interactive learning. The classifier/training system identifies the region in pattern space where images are ambiguous (e.g., the boundary separating categories “l” from “1”), and asks contributors to provide labels of such patterns. In this way interactive learning “focusses in” on difficult patterns, much as in the machine learning technique of boosting [3]. Speeding such interactive learning can be recast as a problem in decision theory in which each action (query) has an expected cost/benefit, and the optimal strategy is to present the query which, when answered, is expected to lead to the greatest improvement in the classifier's accuracy [9, 17]. Below we shall present some techniques based on this decision theoretic framework.

3 Open Mind projects

Three Open Mind projects are currently in development. Open Mind speech recognition addresses speaker identification and the recognition of isolated spoken *Linux* commands [18]. Open Mind common sense builds common sense ontologies and reasoning mechanisms; netizens contribute simple assertions (e.g., “grass is green”), ontology data (“all chairs are furniture”), and abstract inferencing rules [13]. Open Mind handwriting addresses the recognition of handwritten letters and English words [10].

4 Open Mind *Animals*

To explore the issue of ensuring data quality in open data collection we implemented a simple demonstration “20 questions” AI game, *Animals* [11, 16]. A binary tree data structure based on animal attributes is grown during sessions (games) as follows. The player thinks of a target animal, which the system tries to guess based on the player's answers to a sequence of yes/no queries corresponding to a path through the tree (Fig. 1).

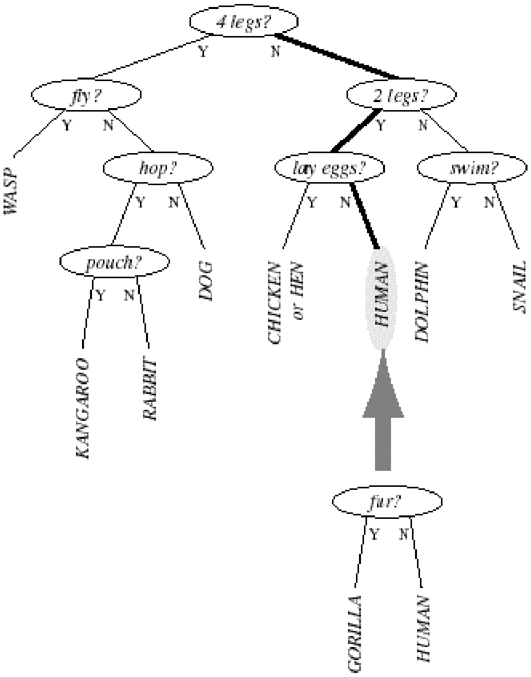

Figure 1: Here the player's target animal is *GORILLA* but the system guesses instead *HUMAN*. Since this guess is incorrect, the player is then asked to contribute a new question that distinguishes her target animal from the one guessed by the system, for instance “*have fur?*”. This new question replaces the previous leaf node and the tree is expanded, as shown.

The system guesses the target animal according to the label at the leaf at the end of the path. If this guess is correct, then the game is over. If instead the guess is incorrect, then the system asks the player for the identity of the target animal and a single yes/no question that distinguishes this animal from the animal guessed by the system. In our Java implementation, answers to yes/no questions are entered by clicking on one of two buttons and animal names are entered in an HTML form via the keyboard. The more netizens play the game, the larger will be the resulting tree data structure.

The Basic *Animals* procedure is shown here.

Procedure: Basic *Animals*

1. Initialize X to be the root node.
2. If X is a leaf node, then skip to step 6.
3. Ask the question at X.
4. If the player answers "yes," then assign the "left" child node to be the new X; otherwise assign the "right" child node to be the new X.
5. Go to step 2.
6. Guess that the target animal is the one listed at X.
7. If the player answers yes, then stop.
8. Ask player to give a question that distinguishes her target animal from the one guessed by the system; expand the tree using this information.
9. Stop or else restart the game by going to step 1.

5 Ensuring data quality in Open Mind *Animals*

The Basic *Animals* procedure above leads to accurate trees if and only if the data contributed by players is accurate. In the general Open Mind approach there are many netizen contributors, each having different expertise and reliability. The overall goal is to encourage reliable contributors and exploit their data while discouraging unreliable contributors and eliminating their data. Thus we now consider sources of data error and corresponding algorithms, implemented as software modules, for reducing the amount of faulty data accepted by the system. It should be stressed that the modules described here seek to prevent faulty data from being accepted, as is particularly appropriate when growing a hierarchical tree data structure. A number of traditional techniques for ensuring database integrity and accuracy can be used *after* the full tree has been grown [8]. A more detailed discussion of the implementation and analysis is provided in [16].

invalid animal name A player might misspell an animal she seeks to contribute. Thus before an animal is added to the tree, a module checks to see if it occurs in a pre-compiled lexicon of valid animal names.

incorrect item If a player feels an animal guess provided by the system does not correspond to her reply to one of the questions, she can, through a simple dialog interface, bring this questionable leaf node to the attention of the domain expert or another player who independently checks that animal. The questionable node is locked and cannot be split until the verification is complete.

inconsistency/non-uniqueness Suppose a player seeks to add an animal that is already assigned to a different leaf node. This implies there must be a query at an non-leaf node that was answered differently by two players — i.e., an inconsistency in the contributed data. To eliminate such an inconsistency, a module automatically presents to the current player a warning message listing the query on the node on which the players disagree. If the current player nevertheless persists in submitting the animal, the animal name is accepted but both (conflicting) leaf nodes are locked until a domain expert or third player arbitrates the disagreement.

submission collisions In deployments accessible by large numbers of netizens such as on the World Wide Web, it becomes probable that two players will try to contribute animals at a particular leaf node *simultaneously.* To accommodate such colliding submissions, a module allows the first netizen to modify the leaf node but locks that node from the second player for 30 seconds.

synonyms Two netizens may wish to use different names for the same animal (e.g., *CHICKEN* and *HEN*). A module allows the second netizen to post such a synonym for an animal already listed at a leaf node.

subset animal Suppose a player's target animal is *MANX* and that the system guesses *CAT*. Technically speaking, this answer is correct. Nevertheless, we would like to capture the information associated with the subcategory *MANX*. A module allows the player to enter a question that distinguishes her target animal, *MANX*, from all other members of the large category, *CAT*.

We implemented the Basic *Animals* procedure and the modules just described in a corporate research intranet of 22 users, representing a roughly equal mix of Netscape Navigator and Internet Explorer browsers. The modules indeed prevented faulty and inconsistent data from appearing in the tree database, as described more fully in [16].

6 General techniques for ensuring data quality

There are several additional techniques for increasing the proportion of high-quality data contributed to general artificial intelligence and pattern recognition systems through Open Mind.

on-line tutorials and tests One method is to require each netizen to read an on-line tutorial, and to pass an on-line test before he or she can contribute data. Not only would such a tutorial improve the general quality of the information provided, the results of the test would indicate the contributor's reliability or level of expertise. Such test results could therefore control the level of difficulty of problems posed to each contributor, and could be used for arbitrating disagreements between contributors of differing reliabilities.

data voting In data voting, some number k of independently polled contributors must agree on an answer (e.g., the label of a handwritten character) before that data is accepted by the learning system. A large k leads to the acceptance of small, highly reliable data sets; a small k leads to the acceptance of large, less reliable data sets. In this way, the value of k, set by the designer of the learning system, trades data *quality* for data *quantity.* The value k^* which leads to the optimal improvement of the learning system depends on the reliabilities of the contributor population, the difficulty of the task and the current state of the system. In general, the higher the average reliability of the population, the lower the k^*. If a highly unreliable or saboteur player is always notified that her data is rejected because it differs markedly from the consensus, perhaps that player may abstain from further participation.

catch trials A class of problems centers on estimating the reliability of the individual contributors during training. The reliability of a contributor can be estimated by intermittently presenting simple problems having unambiguous answers known by the system. Consider collecting labels for a handwritten OCR system to distinguish "4"s from "9"s. Suppose that with frequency f, a perfectly clear and unambiguous "4" is presented as a query to the contributor. (Such an unambiguous pattern could be selected automatically by the classification system itself or from a set pre-compiled by domain experts.) If the contributor gives an incorrect reply to this pattern, the contributor is judged unreliable, and her previous (cached) data is rejected. A high frequency f of catch trials gives an accurate estimate of the reliability of the contributor but reduces the amount of data on informative patterns. Just as in data voting, the value f^* which leads to the optimal improvement of the system depends on the spectrum of reliabilities of the contributor population, the difficulty of the task and the current state of the system.

searching for boundaries Some simple non-parametric classification techniques, such as the nearest-neighbor algorithm, do not seek to model the category distributions but instead find category *boundaries* directly [4]. Thus, rather than collecting a random sample of patterns in two categories, such approaches seek the set of maximally ambiguous patterns, i.e., ones that are in those categories "equally." One popular method for finding a boundary is to iteratively search along the line in feature space connecting two patterns known to be in different categories. For a given total number of queries to the contributor, a fine search leads to an *accurate* estimation of a *small* number of boundary points while a coarse search gives a *poor* estimate of a *large* number of boundary points. There is an optimal search rate that leads to the fastest improvement in the overall classifier accuracy; this search rate depends on the spectrum of reliabilities of the contributor population, the difficulty of the task and the current state of the system.

7 Future directions

We seem to be moving into an era where increasing the amount of *data*, in conjunction with traditional but slow improvements in domain-specific models, offers the greatest hope of progress for the development of many artificial intelligence and pattern recognition systems. A number of trends, particularly the expansion of the internet and participation of non-experts in web projects, imply that the general area of open data aquisition will be increasingly important in the development of these systems. Indeed, for problems such as general common sense reasoning, it is hard to imagine a cost effective alternative to the Open Mind approach. These trends highlight the need for further research in algorithms for filtering during open data acquisition. They also bode well for opportunities for additional projects in the Open Mind framework.

References

[1] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[2] Mindy Bokser, 1999. Personal communication (Caere Corporation).

[3] Harris Drucker, Corina Cortes, Larry D. Jackel, Yann Le Cun, and Vladimir Vapnik. Boosting and other ensemble methods. *Neural Computation*, 6(6):1289–1301, 1994.

[4] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* Wiley, New York, NY, second edition, 2000.

[5] Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramaswamy Uthurusamy, editors. *Advances in knowledge discovery and data mining.* MIT Press, Cambridge, MA, 1996.

[6] Tin Kam Ho and Henry S. Baird. Large-scale simulation studies in pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-19(10):1067–1079, 1997.

[7] Frederick Jelinek. *Statistical Methods for Speech Recognition.* MIT Press, Cambridge, MA, 1998.

[8] Kamran Parsaye and Mark Chignell. *Intelligent Database Tools and Applications: Hyperinformation access, data quality, visualization, automatic discovery.* Wiley, New York, NY, 1993.

[9] John W. Pratt, Howard Raiffa, and Robert Schlaifer. *Introduction to Statistical Decision Theory.* MIT Press, Cambridge, MA, 1995.

[10] Lambert Schomaker and David G. Stork. Open Mind handwriting recognition, 2000. to be submitted.

[11] Stuart C. Shapiro, January 1982. Programming Project 1: Animal Program (20 Questions), "Introduction to Artificial Intelligence," Department of Computer Science, State University of New York at Buffalo.

[12] Edward H. Shortliffe. *Computer-Based Medical Consultations: MYCIN.* Elsevier/North-Holland, New York, NY, 1976.

[13] Push Singh. Open Mind common sense: A collaborative common sense acquisition system. Technical report, Massachusetts Institute of Technology Media Lab, February 2000.

[14] David G. Stork. Document and character research in the Open Mind Initiative. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR99)*, pages 1–12, Bangalore, India, 1999.

[15] David G. Stork. The Open Mind Initiative. *IEEE Intelligent Systems & their applications*, 14(3):19–20, 1999.

[16] David G. Stork and Chuck Lam. Open Mind *Animals*: An open web-based collaborative framework for growing accurate decision trees, 2000. submitted for publication.

[17] Sebastian Thrun. Exploration in active learning. In Michael Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 381–384, Cambridge, MA, 1995. MIT Press.

[18] Jean-Marc Valin and David G. Stork. Open Mind speech recognition. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU99)*, Keystone, CO, 1999.