Just the Fax–Differentiating Voice and Fax Phone Lines Using Call Billing Data

Haim Kaplan and Martin Strauss and Mario Szegedy
AT&T Labs, Florham Park, NJ 07932 USA
{hkl,mstrauss,ms}@research.att.com

August 7, 1998

Abstract

We survey an experiment in distinguishing voice and fax customers in a long distance telephony network based on the customers' calling patterns. We highlight three elements of algorithmic interest. First, we indicate how the theory of I/O-efficient algorithms improved the performance of our software, whereas other algorithmic formalisms did not. Next, we describe a hyperplane-separator heuristic we developed for this problem whose performance here is comparable to that of an established learning algorithm. Finally, we discuss some open problems whose solution would have direct practical value.

1 Introduction

Each call over a long distance telephony network generates a data record containing, among other things, the origination phone number, termination phone number, start time, and duration of the call. Quick, reliable analysis of this data is increasingly important for such business purposes as fraud detection, detection of usage trends and detection of network congestion or outages.

A long distance telephone network may carry 250-300 million calls in day. Because of this large volume, some algorithms that traditionally have been used to analyze smaller datasets, when applied to phone data, will require more than the available resources of time (spent both computing and fetching data from disk to main memory) and space (both core and disk space). In the experiment we describe, we needed to address these issues.

Among the users of the same long distance network are facsimile users and traditional voice users. Of course, a user may plug a fax machine into a voice line at any time without notifying the service provider. The service provider would want to know (with significant confidence) which users are voice users and which are fax users for all of the reasons outlined above:

- Fraud detection. Legitimate fax usage resembles certain fraudulent voice usage, so understanding fax usage is part of fraud detection.
- Usage trends. For example, advertisers may want to adjust their campaigns to reflect increasing overall fax usage.
- Network status. A network overloaded by fax traffic may require different short term and long term fixes then a network overloaded by voice traffic. Faxes and voice calls can be transmitted over channels with very different characteristics (faxes travel in only one direction and can tolerate brief delays better than voice calls).

2 Problem Statement

From a corporate database, we obtained a set of approximately 80,000 telephone numbers self reported as voice and 15,000 self reported as fax. We removed numbers labeled both ways, as well as obviously incorrect numbers.

> Given a collection of 4-tuple call records (caller, callee, start time, duration), and the labeled set of phone numbers, the problem is to produce a hypothesis that classifies phone numbers as voice or fax given only new call records.

We will be more precise below about hypothesis class and goodness of hypothesis.

We assume that we have enough disk space to store all the relevant call records, but that we cannot load them all simultaneously into core memory. (In practice, the more we can process the better. We chose to work with a number N of call records making these assumptions hold.)

3 Theory, Useful and Useless

The theory of I/O-efficient algorithms [2] models disk latency by counting the number of blocks read, even if only one item from each block is needed. A common principle of this theory advocates usefully processing an entire disk block of records when it is read. Indeed, we found it advantageous to sort the call records in order to group related records into single disk blocks.

At first approach, one might guess that fax machines call fax machines and voice machines call voice machines, so what is needed is an algorithm to find the two connected components (or, more realistically, a sparse cut) of a large graph on phone numbers in which arcs are calls. In fact, fax machines are often capable of making voice calls and telemarketers call both voice and fax machines. Also, there are more voice lines than fax lines. In any case, in our dataset we observed that a "fax line" is more likely to make calls with "voice lines" than with other "fax lines." Connected components, reachability and other transitive closures on graphs are clean formulations, but in practice we found local properties to be more useful and reliable.

4 Learning Algorithms

From the call records we created a file of attribute records for each phone number. Attributes include, among others,

- The number of calls made between 9am and 10am
- The tendency to call at the same time each day
- The number of labeled voice and labeled fax lines with which calls are made.

Computing all these attributes is relatively straightforward, though efficiently computing some attributes requires several rounds of sorting of large sets.

We constructed decision trees from the file of phone number attribute records using C4.5 [5], a well-known learning algorithm. We also constructed hyperplane approximate separators, using our own heuristic, which we now describe.

Given two sets of data $X \subseteq \mathbf{R}^d$ and $Y \subseteq \mathbf{R}^d$ (the vectors of attributes for voice and fax lines), we may regard the data as lying in a d-dimensional plane at unit distance from the origin in $\mathbf{R}^{d+1}$. We can then attempt to separate the vectors by a hyperplane P through the origin in $\mathbf{R}^{d+1}$. We measure the success of P by the number $\mathrm{miscl}(X, Y, P)$ of misclassified points,

$$|\{x \in X : \langle x, n_P \rangle \leq 0\}| + |\{y \in Y : \langle y, n_P \rangle \geq 0\}|,$$

where n_P is the normal vector of P.

The separation is *perfect* if $\mathrm{miscl}(X, Y, P) = 0$. The classical perceptron algorithm is a simple, efficient algorithm that finds a perfectly separating hyperplane iff exists [6, 4]. If such a hyperplane does not exist, then finding a P that $\mathrm{miscl}(X, Y, P)$ is minimum is NP-hard. Several algorithms in the literature try to find a hyperplane P such that miscl(X,Y,P) is as small number as possible [3]. We use our own algorithm, which yields good results. The algorithm iteratively finds hyperplanes with increasing performance, such that each local optimization step is a version of line search ([7]) in some random direction.

We create a sequence of hyperplanes $P_0, P_1, \ldots$ recursively, all going through the origin. We start with an arbitrary hyperplane P_0, which goes through the origin. To construct P_i from P_{i-1}, we pick a random vector $r \in \{0,1\}^d$, and we minimize $\mathrm{miscl}(X, Y, P_{i-1}(\lambda))$ over λ, where $P_{i-1}(\lambda)$ is defined as the hyperplane orthogonal to $n_{P_{i-1}} + \lambda r$. We then put $P_i = P_{i-1}(\lambda)$. Since $P_{i-1} = P_i(0)$, it follows that $\mathrm{miscl}(X, Y, P_i)$ is monotone-decreasing in i. We get good results by limiting the number of iterations to 100 (which probably avoids overfitting).

Neither C4.5's decision trees nor our hyperplanes performed well on phone numbers making extremely few calls. Excluding these, both classifiers performed acceptably well and approximately equally well on unlabeled data, but we note that the hyperplanes are much smaller than the decision trees in representation.

5 Open Questions

We flattened the graph structure completely. At the other extreme, there are nice algorithms for computing transitive closures of massive graphs [1]. Are there I/O-efficient ways of capturing local unflattened structure (first and second neighborhoods)?

In practice, total disk space is limited and some data (i.e., arcs) must be discarded the first time they are seen. Without storing many days' data, can one approximate the tendency of a number to call at the same time each day, or the tendency to call the same set of other numbers?

Give bounds on the performance of our hyperplane algorithm (perhaps making assumptions about the distribution of input).

References

[1] J. Abello and A. Buchsbaum and J. Westbrook. A Functional Approach to External Graph Algorithms. *Proc. 6th Euro. Symp. on Algorithms*, 1998, to appear.

[2] A. Aggarwal and J. S. Vitter. The input/output complexity of sorting and related problems. *Communications of the ACM*, 31(9):1116–1127, September 1988.

[3] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon, Oxford, 1995.

[4] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry.* The MIT Press, 1969.

[5] J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[6] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958. Reprinted in *Neurocomputing* (MIT Press, 1988).

[7] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.