The Natural Log

A proposal for a research project on adaptive user interfaces.

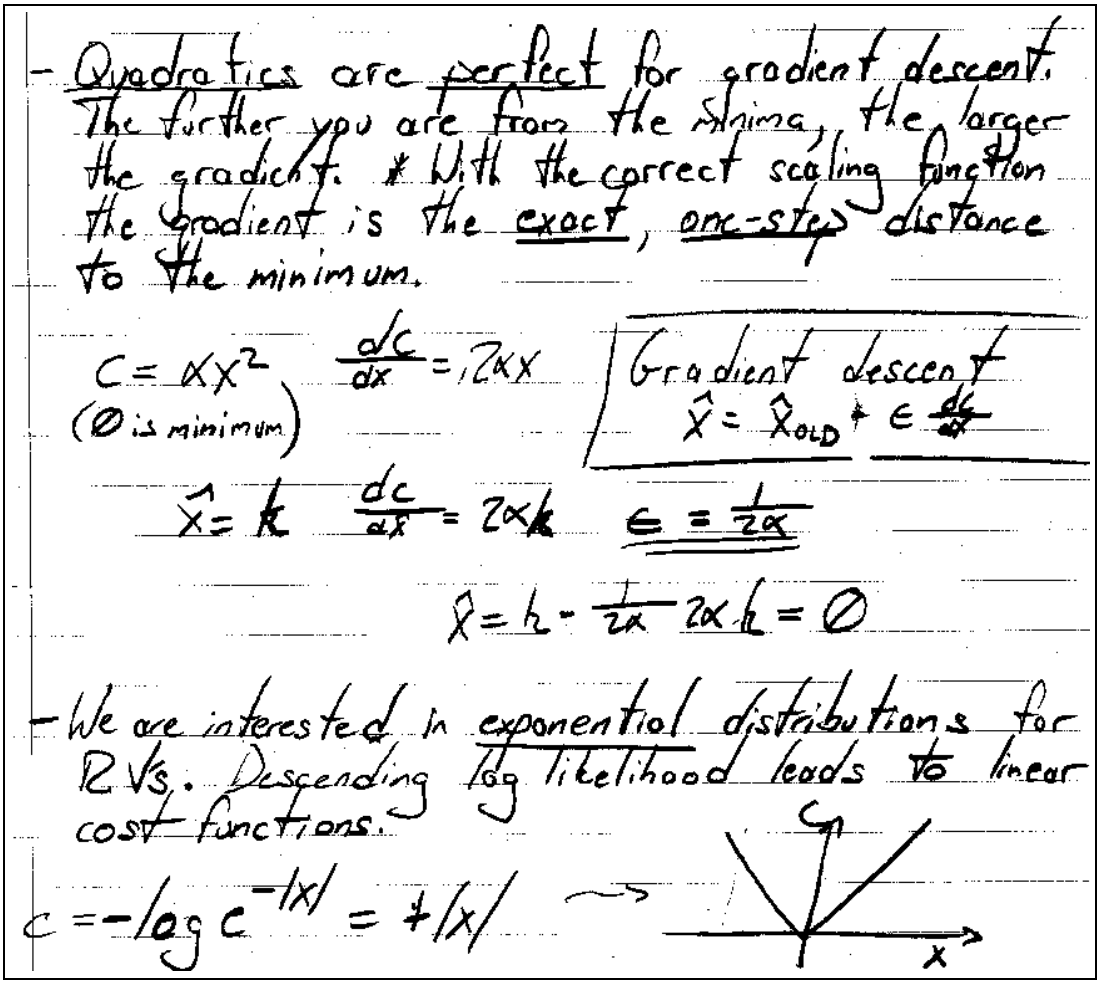
1 Overview

We propose research in two currently distinct areas: i) recognition of potentially ambiguous handwritten text, equations and diagrams and ii) information retrieval from very large databases based on semantic content. Success in these areas will require contributions from a diverse set of disciplines including machine learning, artificial intelligence, neural networks, linguistics, and algorithmic complexity. I have experience in many of these areas and researchers at MIT provide great collaborative potential in others.

In order to focus our research efforts, we propose to construct a new type of intelligent computer interface for use in science and engineering, the *Natural Log*. The Natural Log is a notebook-sized flat display that combines the merits of a technical notebook with the those of a consulting expert. It uses handwriting input to accept text, diagrams, and formulae and combines this with a powerful information retrieval capability that allow access to textbooks, previous work, and tools such as circuit simulators and numerical analysis packages. The Natural Log's greatest value is that it can become an active participant in the users's thought process. As you take notes, the Natural Log reads and interprets your writing. It retrieves relevant documents from an on-line library or from any part of the World Wide Web. You may wish to manipulate a set of equations or tables of data. The Natural Log corrects your algebra, offers simulations of physical or economic systems, or supplies statistical analysis. If you are a student, the Natural Log acts as an interactive tutor who gives you immediate feedback on your work. Based upon what you write or draw it retrieves relevant sections from a wide range of textbooks.

We intend to explore the applicability of the Natural Log to education and instruction in science, mathematics, and engineering where it could be used as a computerized tutor. Clearly the process of solving and then writing up an assignment is an invaluable part of learning. It is our hope that the type of immediate feedback available from the Natural Log will be of equal value. Much like a human tutor, the Natural Log may delay the onset of frustration and provide related, motivating information. We intend to provide access to on-line information from MIT classes including textbooks, videos of classes, assignments, solutions, and previous exams. One important aspect of our research is automatic indexing of such data based on its semantic content. A semantic index allowss the Natural Log to automatically provide information that is applicable to the lesson at hand, and also allow for a free-form exploration of related subjects.

The remainder of this document is divided into three parts: the first section contains our vision for the Natural Log; the second contains a description of the scientific challenges that arise; and the third contains a description of the educational impact of the project.



- Quadratics are perfect for gradient descent. The further you are from the minima, the larger the gradient. * With the correct scaling function the gradient is the exact, one-step distance to the minimum.

$C = \alpha x^2$ $\frac{dc}{dx} = 2\alpha x$
(∅ is minimum)

Gradient descent
$\hat{x} = \hat{x}_{OLD} + \epsilon \frac{dc}{dx}$

$\hat{x} = k$ $\frac{dc}{dx} = 2\alpha k$ $\epsilon = \frac{1}{2\alpha}$

$\hat{x} = k - \frac{1}{2\alpha} 2\alpha k = \emptyset$

- We are interested in exponential distributions for RVs. Descending log likelihood leads to linear cost functions.

$c = -\log e^{-|x|} = +|x|$ $\rightsquigarrow$

Figure 1: An example page from an actual research notebook.

2 Scenario

We expect the Natural Log to be useful in education, engineering, science, finance, management, and in a variety of normal everyday situations. In order to better describe the potential scope of the Natural Log, let us analyze a section from an actual research notebook – Figure 1 shows a scanned section. On this page the student is attempting to more fully understand the notion of function optimization using gradient descent. He starts out by writing something of a conjecture: that gradient descent is the "perfect" technique when the function to be minimized is quadratic. The student then goes on to define a particular function $c(x) = \alpha x^2$ and attempts to show that his conjecture is correct. His next step is to compute the gradient (or derivative) of $c(x)$. Notice for clarity he has stated, and enclosed in a box, the definition of gradient descent. He then shows that a single step of gradient descent will take a sub-optimal solution $\hat{x} = k$ to the optimal solution $\hat{x} = 0$. The student then goes on to analyze a similar situation where the cost function is the log of an exponential. [1] In order to get a feeling for the problem he draws a simple qualitative graph of the new cost function.

The thought process reflected in this notebook is useful for making progress on research problems: state the problem clearly, write out what is known, start with a simple example. Frequently we are asked to perform similar analyses in undergraduate classes (e.g. "Prove that gradient descent is an efficient technique for optimizing quadratic functions."). In this case the act of writing out the question, stating assumptions and then manipulating equations is critical for learning. As an assignment, a problem such as this is not considered difficult because the student has been given all of the information he might need: the problem, related definitions, and possible approaches. Imagine instead that the student is simply exploring, doing research or working on a new type of problem. Since the guidance of the instructor is absent, much of the information which makes this problem easy is now missing. This need not be so.

If the student/researcher were to use a system like the Natural Log, critical related information would always be easily accessible. For example, as the student writes out the initial problem statement, the Natural Log attempts to recognize the text. The text is then used to query on-line resources. For example, the initial statement could be used to retrieve information about quadratic functions, function minimization, or the definition of gradient.

As the student writes out the equation $c = \alpha x^2$, the Natural Log will not only recognize the characters but also determine that this is an equation. The interpretation of hand-drawn equations is difficult because the semantics arise from the geometric relationships between component characters. For example, the "2" is an exponent because it is smaller, higher and to the right of the "x". Once the geometry of an equation is interpreted, the Natural Log can simplify algebra, correct errors, or compute derivatives using the functionality in tools such as Maple, Mathematica, or Matlab.

Finally at the bottom of the page we see that the student has sketched a graph of the equation $c = |x|$. While this is a simple type of plot, other graphs might be much more complex. Whenever a graph of a function is required, the Natural Log could be used to automatically generate one. A user need only roughly sketch the axes of the graph, select an equation and the graph would appear.

This example scenario is a simple but powerful demonstration of the types of facilities we can easily foresee including in the Natural Log. There are several other very different types of user interactions which we intend to support. For example, at the MIT Artificial Intelligence Laboratory Stahovich, Davis, and Shrobe have worked on a system that can interpret drawings of mechanical devices, understand their function and suggest design changes (Stahovich, Davis and Shrobe, 1996). Focusing mostly on higher level reasoning, their system did not use "natural" drawing but instead used a special type of drawing program in which there was little or no ambiguity. We hope to collaborate with them to build a more natural tool for mechanical device design.

[1] This type of cost functions appear frequently in statistical analysis, but the details do not matter here.

Along similar lines Koile has constructed a system that can analyze architectural designs of homes for properties such as “privacy” (Koile, 1997). Her system lacks user-friendly interface and each plan must be entered in a painstaking fashion. Architects typically design buildings by producing a series of rough sketches. In collaboration with Koile we hope to use the Natural Log as a front-end so that sketches of a floor plan might be analyzed early in the design process.

3 Proposed Research Areas

What prevents us from constructing the Natural Log today? The simplest answer is that computers are not intelligent enough to do the task. Successful software exists that can recognize handwritten words and characters. There has been less work on the automatic recognition of diagrams and equations (see (Saund and Moran, 1995) for example). Most current systems are either confused by or ignore diagrams and equations. User input can be “cleaned up” so that the lines are straight, but a diagram representing a mechanical device or electrical circuit remains just a collection of lines to a computer. In order to move to complex automatic analysis, it is necessary that our computers not only have have the ability to understand written text but the ability to extract semantic meaning from drawings. Construction of the Natural Log will be no easy task. The successful system will require two critical components: handwritten text, equation and diagram recognition and a comprehensive natural language retrieval system for on-line textbooks, articles and the web.

3.1 Handwritten Input

It is our goal to construct a system that will accept unconstrained handwritten text. There has been a great deal of research in this area, and many commercial systems are available. Research has been divided on the basis of input: pen-based or optical. Pen-based systems have attained high recognition rates from natural handwriting—the commercial system available with the new Apple Newton is especially impressive (Yaeger, Lyon and Webb, 1995). Optical systems, in part because they have less information, are not as successful with unconstrained input. They have, however, achieved useful levels of performance in limited domains such as the recognition of handwritten addresses on envelopes (Palumbo and Srihari, 1996).

We have begun work on the recognition of handwriting using an approach which is a synthesis of techniques from character recognition and vision. Simard et al. proposed a system for optical digit recognition that could learn automatically from examples and could easily take advantage of a priori information (Simard, Le Cun and Denker, 1993). Their system is a variant of nearest neighbor classification, where a character is recognized by finding the most similar example from the training set. These systems are quite simple, and reasonably efficient, but they require the user to define a metric for the comparison of characters. The most widely used metric is the sum of the squares of the differences between pixels. It this is equivalent to treating each character as a single vector in a high dimensional “character space”, where the

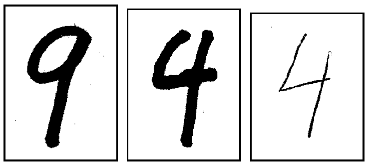


Figure 2: One type of inconsequential difference is pen width. Under a vector difference model the thick four looks more like a thick nine than a thin four.

components of the vector are the values of the pixels in the image. The sum of squares of the differences between pixels is then simply the squared length of the difference vector between two characters.

Vector distance is usually a poor choice for classifying characters. Many changes in a character do not effect its identity, like scale, small rotations, and the thickness of the pen used to write the character (see Figure 2). But sum of squared differences does not differentiate between inconsequential differences such as these and important differences. Simard's insight was to take the vector difference between two characters and to project it onto a sub-space where only *consequential* differences remain. The distance in this sub-space is the tangent distance[2].

The tangent distance is quite similar to the use of the Mahalanobis distance for classification (Duda and Hart, 1973). The Mahalanobis distance is non-isometric, shrinking distance in some axes while stretching it in others. These axes are determined by fitting a gaussian distribution to the data. In contrast, for the tangent distances these axes are computed from the characters themselves using *a priori* information. Both the Mahalanobis and the tangent distance are linear techniques and as such they are only valid when the set of inconsequential variation is also linear. Since many inconsequential variations are non-linear in image space, both approaches are only approximate.

Unlike approaches that measure images, the Natural Log is a pen-based system which measures a *trajectory* of pen positions versus time. This representation is called a stroke. Excluding velocity, the two representations for a character are almost equivalent: a stroke can be converted to an image, and in many cases an image can be converted to a stroke. But these representations are quite different mathematically. The manifold of strokes which have been scaled, rotated or translated is *linear* in stroke space[3]. As a result, the linearity of the Mahalanobis and tangent distance is not a drawback.

We use a combination of Mahalanobis distance and Tangent distance to recognize strokes. In other words some of the axes of variation are determined using a priori information like rotation, scale, and translation. Other axes are determined by fitting a gaussian to a collection of characters (these axes are the eigenvectors of the covariance matrix of the data). This approach has allowed us to build a recognizer for twenty five different characters (see Figure 3 for some of the characters). One

[2]Simard et al.'s derivation of this insight is more complex than this, but their results can be reinterpreted in this fashion.

[3]This was originally observed by Ullman and Basri in the context of 3D object models (Ullman and Basri, 1989).

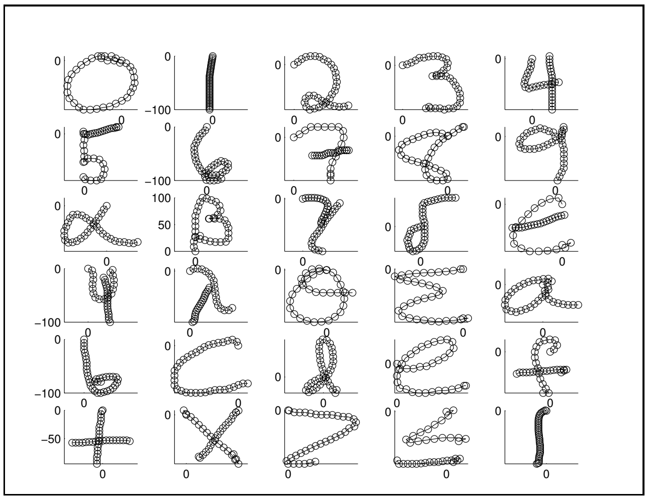


Figure 3: Examples of the different characters recognized by our demonstration program.

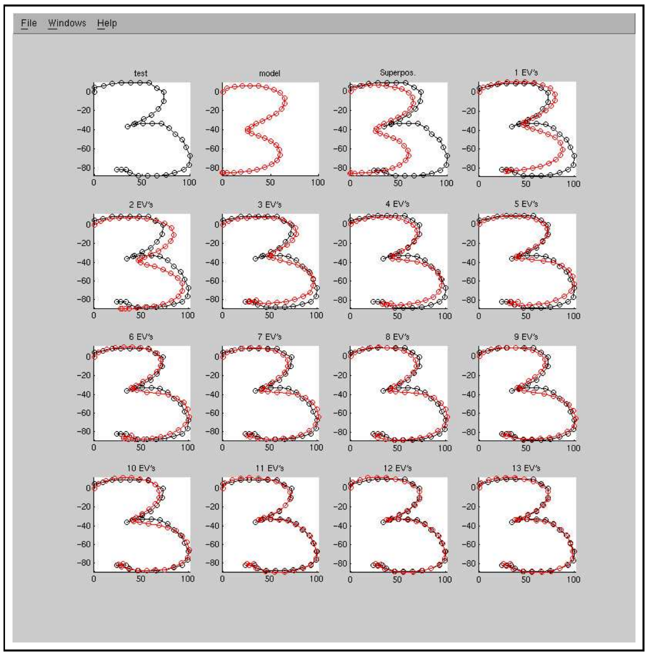


Figure 4: A comparison of two different "3"'s. At the upper left is the raw input. Moving to the right and down axes of variability are sequentially removed. Notice that in the final image the two characters are almost identical.

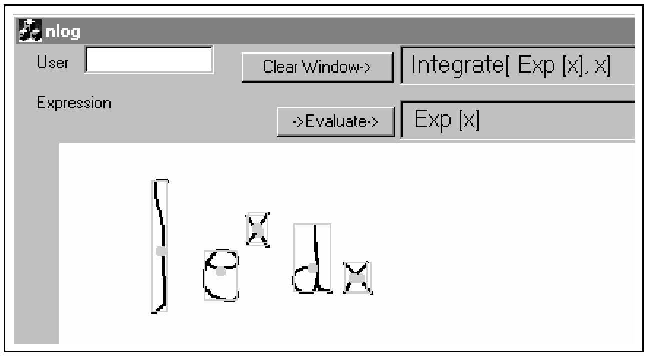


Figure 5: A snapshot of the equation demonstration program. The user has just completed an equation. The box at the top right shows the current interpretation: "Integrate[Exp[x], x]". The box just below that shows one possible simplification: "Exp[x]".

way to visualize the approach is to look at the difference between characters after the allowed axes of variation have been removed. Figure 4 shows a series of comparisons between two "3"'s. In the upper left are the two strokes in their original form. In each subsequent figure one additional axis of allowed variation has been removed. Notice that that eventually the two characters appear almost identical.

The character recognizer works in real-time and has been used as part of a demonstration system, NLOG, which recognizes equations (see Figure 5). In NLOG users can write out very simple equations using twenty five symbols including digits, some letters, and a number of mathematical operators. The program then recognizes and parses the expression where possible (not every type of equation is currently supported). The resulting equation can be sent to a Mathematica kernel that performs algebraic simplification.

3.1.1 Scope of Future Research

Our research in the area of handwriting interpretation is in its early stages. Not surprisingly the current demo has limited functionality and makes a large number of errors. Nevertheless it has pointed out a number of issues for future research:

Isolated Character Recognition: Isolated character recognition performance is currently about 90% for writers for whom the system was trained—perhaps only 75% percent for new writers. One major focus for our research will be improving this performance. Currently the system models each character as a single "average" character plus allowed axes of variation. Since not all writers form characters in a similar fashion, performance should improve if we include multiple models for each character. Our approach for finding multiple models is statistical. Multiple models are necessary when a single model is unable to explain an entire set of training characters (recent work by (Tipping and Bishop, 1997) provides a theoretical mechanism for this inference).

Statistical Models for Disambiguation: Some characters will never be easily distinguished, for example a 'zero' and the letter 'o'. Both speech and handwriting systems use statistical language models to differentiate such characters. For example, these models can be used to rule out the possibility that the 'o' in "for" is a 'zero'. Language models are typically extensions of Markov Models for the production of words and characters (see (Brown et al., 1992) for example). We hope to build similar statistical models for other types of handwritten input like equations, and diagrams. There is one type of immediate difficulty that arises when applying such models: while Markov models are tractable for linear sequences, like language, they are intractable for more complex spatial structures. Inference using Markov models for two dimensional images is well known to be computationally expensive (Geman and Geman, 1984). We have recently proposed an efficient mechanism for performing statistical inference on two dimensional images (De Bonet and Viola, 1997a). One measure of the quality of a statistical model is to draw samples from that model. A good language model will generate samples that look like English text. A good image model should generate images that look natural. Our system can generate images that are much more realistic than previous approaches[4]. We will extend this model so that it can be used on the geometric distributions that appear in equations and diagrams.

Contextual Reasoning: In the final analysis many types of equations and diagrams may remain ambiguous, such as "$sin(x)$". Is this the function $sin()$ applied to the argument x? Or is it the product $s * i * n(x)$? In many cases statistical models can be used to decide on the correct interpretation—$sin(x)$ may be more likely than $s * i * n(x)$. But such a model cannot be an unchanging function of time. The correct interpretation depends on the local context. If the previous few equations involved the variables s, i and x and the function $n()$, then the correct interpretation is likely to be different than if the previous lines referred to $cos(x)$ and $tan(x)$. We will construct a representation for this contextual information so that it can be used by our statistical models. Optimal recognition performance then becomes a matter of retrieving the correct contextual model for entering particular equations or diagrams.

3.2 Semantic Indices for Very Large Databases

After text, equations, and diagrams have been interpreted, the Natural Log will be able to retrieve relevant information from a large database of textbooks, articles, and the World Wide Web. This information need not be entirely text-based. The Natural Log will also be able to retrieve applicable computational and analysis tools such as circuit simulators, statistical tests, and algebraic manipulation tools. This information will be retrieved automatically (or perhaps semi-automatically) based on the contextual information.

Naturally, we would like to apply techniques from natural language understanding to this problem. Unfortunately, the sheer size of the data to be represented makes this difficult. At a bare minimum, we wish to process 100,000 documents, each of which

[4] See `http://www.ai.mit.edu/people/jsd/jsd.doit/Research/TextureSynthesis` for many example syntheses.

may contain any of up to 10,000 different words. Current natural language systems require significant computation and would have difficulty processing this much data.

3.2.1 The Vector Space Model

Our current work attempts to improve upon Latent Semantic Indexing (LSI), a technique that uses The Vector Space Model (VSM) (Salton, 1971) as a representational basis. In VSM, a document is represented as a vector where each component is a count of occurrences of a particular word (Salton, 1971)[5]. A collection of documents is a matrix, D, where each column is a document vector d_i. Queries are similarly represented, making queries nothing more than short documents. The similarity between two documents, d_i and d_j, is defined to be their inner product, $d_i^T d_j$, so scoring the relevance of documents to a query, q, reduces to computing $D^T q$. This is essentially the same approach used by many search engines, from library systems commonly available in universities to the wildly popular Alta Vista web search engine.

There are several advantages to this approach beyond its mathematical simplicity. Above all, it is efficient to compute and store the document matrix. The document matrix will be sparse, because most documents will contain only a few of the 100,000 possible words (i.e. most of the word counts will be zero). Thus algorithms for manipulating the matrix are quite efficient. Nevertheless, VSM makes an important tradeoff by sacrificing a great deal of document structure. Specifically, VSM removes information about the order and proximity of terms in a document, losing context that may disambiguate meaning. LSI is one common method for improving retrieval performance.

LSI modifies the document matrix so that it retains only the most important information from the original. In particular, LSI uses the singular value decomposition of the document matrix $D = USV^T$ where U and V contain orthogonal vectors and S is diagonal (see (Golub and Loan, 1993) for further properties and algorithms). From S and U we can define $\hat{S}_k$ and $\hat{U}_k$ which contain only the largest k singular values and corresponding eigenvectors, respectively. In LSI, each document is projected into a lower dimensional space $\hat{D} = \hat{S}_k^{-1} \hat{U}_k^T D$. The resulting document matrix $\hat{D}$ is of smaller size but still provably represents the most variation in the original matrix. In other words, LSI represents documents as a linear combination of a few orthogonal "features". It is hoped that these features represent meaningful underlying topics present in the collection. Queries are also projected into this space. Scoring the relevance of documents to a query is therefore $D^T \hat{U}_k \hat{S}_k^{-2} \hat{U}_k^T q$.

Though it has resisted a formal justification, experiments have shown that projection onto $\hat{U}_k$ does improve retrieval performance. Hypotheses abound, including: i) LSI removes noise from the document set; ii) LSI finds words that are synonyms; iii) LSI finds clusters of documents. Whatever it does, LSI operates in a completely unsupervised fashion without knowledge of the queries that may be presented to the system.

[5] In practice, suffixes are removed and counts are weighted by their natural frequency (Frakes and Baeza-Yates, 1992). This is known to improve performance significantly (we have incorporated these methods); however, such details are unimportant for this discussion.

3.2.2 The Optimal Projection Matrix

We could instead attempt a supervised approach, where we are given a collection of documents, D, a collection of queries, Q, and a relevance matrix R such that $R_{ij} = 1$ if document i is relevant to query j and 0 otherwise. We can use this information to construct an optimal P such that: $D^T P P^T Q \approx R$. We find P in two steps: first we find an X minimizing $\|D^T X Q - R\|_F$, where $\| \cdot \|_F$ denotes the Frobenius norm of a matrix, then we can attempt to find P by decomposing X into PP^T. Note that X is the optimal linear transformation of the documents so that when they are compared to the queries only relevant documents are retrieved. Since there is no better *linear* transformation it can act as a benchmark for the VSM retrieval approach.

Computation of P is not a simple matter. The matrix PP^T is necessarily symmetric and positive semidefinite. There is no guarantee that X will satisfy any of these properties. In fact, for the Medline[6] document collection X is far from symmetric. When X is not symmetric it does not project the documents and the queries in the same way (as was assumed above). In order to understand how X acts on documents and queries we choose to decompose it into two orthogonal projection matrices with SVD, $X = U_X S_X V_X^T$. The projection matrix for for D is $U_X S_X^{1/2}$ and the projection matrix for Q is $V_X S_X^{1/2}$.

We can examine the "optimal" projection axes to determine whether they are similar to the axes selected by LSI. Figure 6a shows the distribution of Medline documents projected onto the first axis of U_X. Notice that there is a large spike near zero, and a well-separated outlier spike. Subsequent axes of U_X result in similar distributions. One way to interpret such axes are as small clusters of documents: most of the documents have close to zero dot product and a small cluster of documents have a large dot product. It is possible that this axis might correspond to a topic shared by a few documents. In contrast, if one were to project the data onto a randomly selected axis the resulting distribution is almost gaussian.[7]. This can be quantified by measuring the kurtosis of the distribution. For the medline data the optimal axes yield a kurtosis between 20 and 50 while the kurtosis of a gaussian is 3. LSI axes yield distributions which are much closer to gaussian. Figure 6b shows the distribution of documents projected onto the first LSI axis. This axis yields a distribution with a much lower kurtosis of 6.9. This axis is nothing like a cluster of documents: there is a continuum of dot product values.

3.2.3 The TOPIC Representation

These results strongly suggest that LSI may well be finding projections which are quite different from the optimal linear projection. Unfortunately we cannot use the optimal axes, because they are computed from information which we do not have–a list of documents relevant for every query. An alternative might be to search for projection axes with properties like those of U_X. We conjecture that axes corresponding to

[6] The Medline document collection contains over 1000 documents. It includes thirty queries with indications of which documents have been manually judged relevant to each query.

[7] This is predicted by the theoretical analysis of (Diaconis and Freedman, 1984).

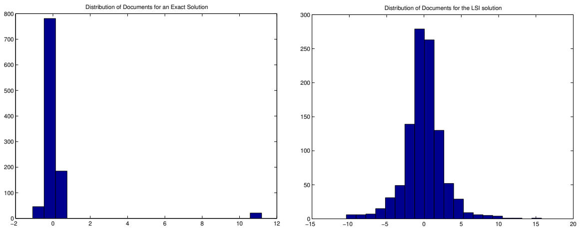


Figure 6: **(A).** Projection of the documents onto one of the "optimal" linear axes. The resulting distribution has a very high kurtosis (44). **(B).** Projection of the documents onto one of the LSI axes. The resulting distribution has a moderate kurtosis of 6.9.

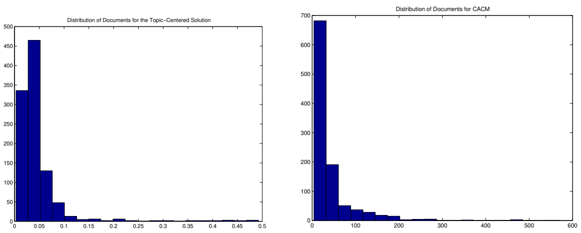


Figure 7: **(A).** The distribution of Medline documents when projected onto a TOPIC axis (kurtosis=43). **(B).** The distribution of CACM documents (containing 3200 documents and about fifty instrumented queries) using TOPIC (kurtosis=25).

a cluster of documents will yield good performance. We have experimented with a number of clustering algorithms but the following simple algorithm performs equally well:

```
for   each axis you wish to define
      Choose a source document
      Choose a small number, k, of nearest neighbors
      Use the centroid of those documents as a projection axis
```

We then construct a projection matrix from these vectors, C. For retrieval, both the document matrix and the query are projected into this space $D^T C C^T q$. We call this TOPIC based retrieval because documents and queries are represented not as a vector of words but instead as a vector of "topics". The properties of the topic axes are much closer to the optimal U_X than they are to $\hat{U}_k$. Figure 7 shows how the collection of documents project onto typical TOPIC vectors.

TOPIC based retrieval has several distinct advantages over LSI. TOPIC is efficient to compute, while LSI requires a much more complex eigenvector computation. TOPIC preserves the sparse nature of the document matrix, while LSI does not. The LSI document matrix, $\hat{D} = \hat{S}_k^{-1} \hat{U}_k^T D$ has many fewer rows than D but it often contains far more non-zero entries. Most of the entries in TOPIC document matrix $C^T D$ are close to zero. Setting them to zero does not reduce retrieval performance. Finally, in several experiments TOPIC has outperformed LSI on retrieval of relevant documents. Figures 8 and Figure 9 show precision-recall curves for LSI and TOPIC[8]

[8]Following convention, we measure the success of a text retrieval system using precision-recall

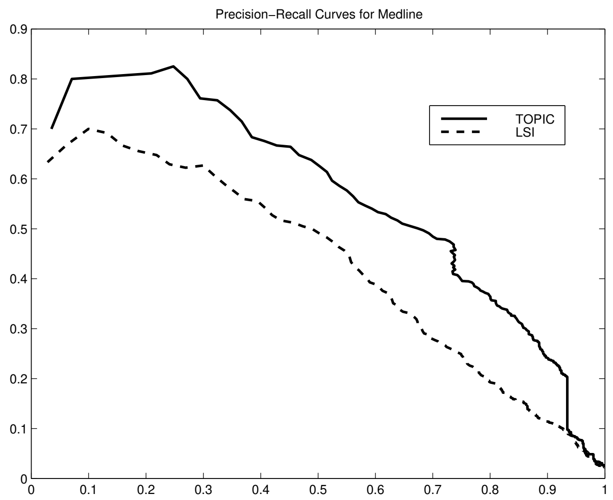


Figure 8: A comparison of TOPIC and LSI on Medline. TOPIC performs better than LSI in almost all cases.

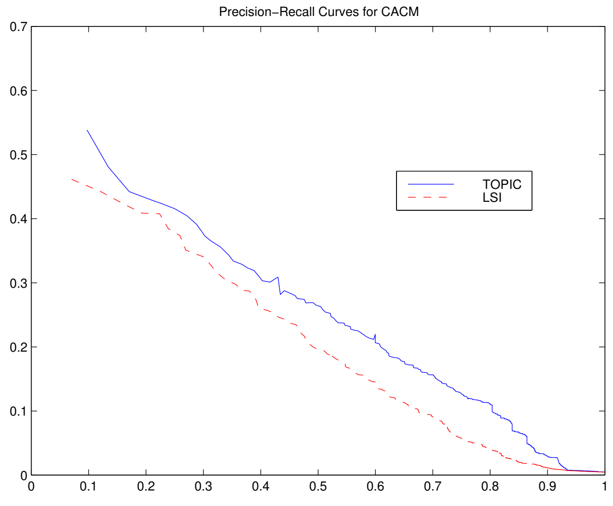


Figure 9: A comparison of TOPIC and LSI on CACM.

3.2.4 Scope of Future Research

This work has pointed out that representing documents as collections of topics is better than representing them as collections of words. Nevertheless, performance is still far from perfect and is in fact just barely acceptable. We intend to explore several avenues in an effort to achieve a significant improvement in performance.

Formal Analysis: Our current thinking, like much work in information retrieval, is empirical. We will study the vector space model with an eye toward a theoretical justification both of LSI and TOPIC. We believe that a theory based on information transmission will provide a justification for TOPIC. Such theories have been influential in signal and image processing (Bell and Sejnowski, 1995; Viola, Schraudolph and Sejnowski, 1995).

More Complex Features: Potential for improving retrieval using the Vector Space Model is almost certainly limited. For example, a VSM representation can never understand that "National Football League" has a special unique meaning that is not entirely a function of its component words. One possibility is to augment the VSM representation to include more complex features constructed from combinations of words or based on parts of speech. One immediate difficulty arises from the explosion of possible features. While there are roughly 100,000 different words in a large document database, there is an astronomically large number of word pairs. We must somehow select from among these potential features efficiently. We have had some initial success extracting multi-word nouns such as "Euler's number" and "differential equation" from documents as additional features. We will also explore the role of part-of-speech tags.

Extension to Other Types of Documents: We have recently constructed a system that performs image database retrieval. Like text document retrieval, good image retrieval requires the interpretation of semantic content. Our approach, which is reminiscent of VSM, constructs very high dimensional vectors of features (De Bonet and Viola, 1997b)[9] Retrieval of documents is performed as in VSM, using a dot product. We intend to explore the role of TOPIC in image database applications.

We will explore the applicability of our techniques to other types of data like programs and theorems.

Document Compression: One of the largest problems in retrieval is the huge size of document databases (we have done some experiments with a 3 Gigabyte database). If it were possible to compress the text, then part of this difficulty would be removed. It is important however that these compressed representations

---

curves. Given a query, an text retrieval system ranks each document in a collection. If we restrict the system to return only its top n documents, we can determine the proportion of those n documents returned that are actually relevant. This is *precision*. We can also compute what proportion of the total number of relevant documents existing in the database were actually returned. This is *recall*. For each n, the average precision and recall over all of the queries is computed and precision plotted against recall. On these graphs, a curve which is higher implies that more relevant documents are retrieved out of a group.

[9] See `http://www.ai.mit.edu/people/jsd/jsd.doit/Research/ImageDatabase/Demo` for a demonstration of the system.

be just as easily compared and retrieved as the original text. Recently de Marken has achieved high compression rates by removing the redundancies in natural language texts (de Marcken, 1996). The representation that his system constructs reflects the structure of language and as such is useful for retrieval. We plan to collaborate with de Marken in this area.

Language Modeling: Progress in natural language understanding will have great impact on document retrieval. Nevertheless because of computational complexity issues we do not believe natural language understanding will be possible on huge databases. Potentially these systems could however determine that documents are closely related even though they share few words. This information could be used in the construction of TOPIC vectors which could then be used in more efficient vector-based techniques.

Fine Semantic Discrimination: The different chapters of a textbook on Differential Equations are all about the same global topic, but each describes a different sub-topic. Current retrieval techniques are overwhelmed by the similarities between these documents and find it difficult to discover the fine distinctions. We will explore hierarchical representations for topics that will expose difference that would be otherwise overlooked.

4 Impact on Education

This project will directly impact the education of students at MIT in many ways:

Graduate Education: The Natural Log has already attracted the interest of two graduate students who are currently supported with other funds. These students have helped significantly in the conception and early implementation of the handwriting recognition and the natural language document retrieval systems. The goals of the Natural Log has helped them to focus on concrete results and evaluation metrics. But more importantly the breadth of the Natural Log has opened their thinking to other areas of Artificial Intelligence like reasoning and machine learning.

Undergraduate Education: The Natural Log has been an effective motivational tool for undergraduates. MIT has two programs which involve undergraduates with research: the undergraduate research opportunities program (UROP) and the undergraduate thesis. Since the conception of the project six months ago I have had several undergraduates work with me. They have immediately responded to the exciting potential of the Natural Log and have contributed long hours reading papers or writing code. We meet weekly and the students have surprised me with their accomplishments. Based on this positive response, I will attempt to initiate a small seminar focusing on the science and technology in the Natural Log.

Teaching Aid: Throughout the conception of the Natural Log we have envisioned it as aid both for research and learning. The same tools that will help a researcher solve an unsolved problem can help a student discover the solution to an assigned problem. In the final years of this grant we will have a system that can be tested as an educational tool. Though it is tempting to predict that the system will be immediately useful for students, we expect there to be significant additional work required. We will search out collaborative assistance with issues in course design and pedagogy.

5 Relationship to Commercial Endeavors

There has been much commercial interest in pen-based hand-held computers such as the Newton or the Pilot. These devices are pioneering attempts in the effort to build small computers. But currently there is no commercial software that attempts to play the role of the Natural Log. Nevertheless we expect to benefit from technological advances fostered by these and related products. For example we intend to build as little hardware as possible. Our focus is on artificial intelligence and machine learning done with software. The results of our work will be general purpose algorithms and insights, not hardware. We will purchase the best available pen based computers and where possible use “off the shelf” software. For example we have evaluated all of the commercially available handwriting recognition software for use in the Natural Log and none was flexible enough for our purposes. We are currently using commercial mathematical and simulation software such as Matlab, Mathematica and Spice.

6 Conclusions

A successful approach to constructing the Natural Log must combine advances in machine learning, pattern recognition, linguistics, and algorithms for ultra-large databases. Principled solutions to the technical challenges that we have identified will lead to a very general facility for generating different types of Natural Logs for many domains. For example, it should be possible to create a system to assist a corporate financial officer who must make a recommendation for an investment. In this case, the text that would be indexed would be financial and the operations available would focus on the estimation of yield and risk.

Preliminary work on the Natural Log has begun. We are actively studying handwriting recognition, both of text and figures. We have also begun work on the semantic retrieval of text documents. Though the task ahead is a large one, we are confident that the Natural Log will eventually become an invaluable tool for the student, scientist or engineer.

References

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479. learning language IBM.

De Bonet, J. and Viola, P. (1997a). A non-parametric multi-scale statistical model for natural images. Submitted to Advances in Neural Information Processing 1997.

De Bonet, J. and Viola, P. (1997b). Structure driven image database retrieval. Submitted to Advances in Neural Information Processing 1997.

de Marcken, C. (1996). *Unsupervised Language Acquisition*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Diaconis, P. and Freedman, D. (1984). Asymptotics of graphical projection pursuit. *Annals of Statistics*, 12:793–815.

Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons.

Frakes, W. B. and Baeza-Yates, R., editors (1992). *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions PAMI*, PAMI-6(6):721–741.

Golub, G. H. and Loan, C. F. V. (1993). *Matrix Computations*. The Johns Hopkins University Press.

Koile, K. (1997). Design conversation with your computer: Evaluating experiential qualities of physical form. to appear in CAAD Futures '97, Proceedings of the Seventh International Conference on Computer-Aided Architectural Design Futures, Aug 3-7, 97, Munich, Germany.

Kwok, K. L. (1996). A new method of weighting query terms for ad-hoc retrieval. In *Proceedings of the 19th ACM/SIGIR Conference*, pages 187–195.

O'Brien, G. W. (1994). Information management tools for updating an svd-encoded indexing scheme. Technical Report UT-CS-94-259, University of Tennessee.

Palumbo, P. and Srihari, S. (1996). Postal address reading in real time. *International Journal of Imaging Science and Technology*.

Salton, G., editor (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall.

Saund, E. and Moran, T. (1995). Perceptual organization in an interactive sketch editing application. In *Proceedings of the International Conference on Computer Vision*, Cambridge, MA. IEEE, Washington, DC.

Simard, P., Le Cun, Y., and Denker, J. (1993). Efficient pattern recognition using a new transformation distance. In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing*, volume 5, pages 50–58, Denver 1992. Morgan Kaufmann, San Mateo.

Stahovich, T. F., Davis, R., and Shrobe, H. (1996). Generating multiple new designs from a sketch. In *Proceedings Thirteenth National Conference on Artificial Intelligence*, pages 1022–29. Proceedings AAAI.

Tipping, M. E. and Bishop, C. M. (1997). Mixtures of probabilistic principal component analysers. Technical Report TR NCRG/97/003, Neural Computing Research Group, Dept of CS and Applied Math, Aston Univ, Birmingham, England.

Ullman, S. and Basri, R. (1989). Recognition by Linear Combinations of Models. A.I. Memo 1152, Massachusetts Institute of Technology.

Viola, P. A., Schraudolph, N. N., and Sejnowski, T. J. (1995). Empirical entropy manipulation for real-world problems. In David S. Touretzky, M. M. and Perrone, M., editors, *Advances in Neural Information Processing*, volume 8, Denver 1995. MIT Press, Cambridge.

Yaeger, L., Lyon, R., and Webb, B. (1995). Effective training of a neural network character classifier for word recognition. In Mozer, M., Jordan, M., and Petsche, T., editors, *Advances in Neural Information Processing*, volume 9, Denver 1996. MIT Press, Cambridge.