Hybrid Hill-Climbing and Knowledge-Based Methods for Intelligent News Filtering

Kenrick J. Mock

Department of Computer Science
University of California Davis, Davis, CA 95616
Now at Intel Corporation, JF2-76
2111 N.E. 25th Avenue, Hillsboro, OR 97124
Kenrick_J_Mock@ccm.jf.intel.com

Abstract

As the size of the Internet increases, the amount of data available to users has dramatically risen, resulting in an information overload for users. This work involved the creation of an intelligent information news filtering system named INFOS (Intelligent News Filtering Organizational System) to reduce the user's search burden by automatically eliminating Usenet news articles predicted to be irrelevant. These predictions are learned automatically by adapting an internal user model that is based upon features taken from articles and collaborative features derived from other users. The features are manipulated through keyword-based techniques and knowledge-based techniques to perform the actual filtering. Knowledge-based systems have the advantage of analyzing input text in detail, but at the cost of computational complexity and the difficulty of scaling up to large domains. In contrast, statistical and keyword approaches scale up readily but result in a shallower understanding of the input. A hybrid system integrating both approaches improves accuracy over keyword approaches, supports domain knowledge, and retains scalability. The system would be enhanced by more robust word disambiguation.

The Information Overload Problem

The goal of this project is to predict whether new news articles are likely to be of interest, or not of interest, based upon the prior behavior of the user. Systems that perform this type of intelligent behavior have recently been touted as intelligent "agents" by the media. The work proposed here follows the same vein; the system is intended to aid the user in her work and to learn what the user is interested in so that intelligent filtering may be performed. The filtering task is an extremely fuzzy and difficult problem to solve since users are notorious for their inconsistencies in behavior and interests. From a machine learning perspective, the problem is similar to trying to approximate a curve based upon discrete data points - except in this case, the function may change at any time.

To illustrate the filtering task, consider a stream of news articles consisting of hundreds of articles posted daily. If the reader is interested only in articles concerning Bayesian induction, then all other articles may be considered as noise. Picking only the relevant articles from the news stream is a time-consuming task for humans and is the objective of the filtering system.

Previous Work

Is a news filter even necessary? To answer this question, a study was conducted by Mock (1996) that observed the behavior of users while reading news. Experimental results indicated that existing methods for browsing result in many messages that users do not read, but would be interested in reading. Furthermore, the results indicate that users often change their mind about whether they like or dislike a particular article. A news filter would be a great aid in finding articles likely to be of interest that are normally missed, as well as removing articles that are not of interest. However, the accuracy of such a filter may be limited due to human inconsistencies. Consequently, any filtering system must be flexible and easily user-modifiable in order to minimize error.

Before a news article may be intelligently processed, the article must first be "understood" to some degree by the system. For information filtering, incoming articles must be understood well enough so that the content can be compared with the user model to determine if there is a match. A common assumption has been that articles need not be understood as well as a human reader in order to determine whether or not interest exists.

Typically, understanding is demonstrated by the extraction of key features from the text or by providing a summary of the article. The easiest method of feature extraction is simply to pull *keywords* or *tokens* from the text that match a predefined set of words describing a user's interests or simply to use all of the words in the input article as features (Jennings & Higuchi, 1992). Often, the words are first passed through a *stemmer* and a *stop list.* A stemmer attempts to strip away word prefixes or suffixes to find the word root for comparison purposes. A stop list is a list containing common words that have no predictive value. These words are thrown out entirely. While the keyword/stemmer/stop list approach can be effective, it is difficult to predefine all relevant keywords and synonyms that may occur in a text, or text may be worded in a manner that does not match a keyword.

After keywords have been scanned from news articles, a popular method of indexing the news document with the extracted terms is to use rule-based agents to model a user's usage patterns as in INFOSCOPE (Stevens, 1992), or to couple the term-frequency with the inverse-document frequency. This method is often referred to as *tf-idf* (Salton, 1991). These two terms are combined by multiplying the term-frequency (*tf*) by 1/document-frequency (*idf*) to obtain a metric of relevancy for each term. By combining terms from a document to form a vector, queries can undergo a similar process and the document vector closest to the query vector is retrieved as the best match.

To use the tf-idf method for information filtering, the tf-idf statistics are collected for an entire class of news articles. A simple two class system might include articles the user is interested in reading, and articles the user is not interested in reading. The similarity of a new article is computed by comparing it against classes instead of against individual articles, and the class most similar to the new article is used to predict the user's interest in the unread article. The NewT system (Sheth, 1994) is based upon tf-idf and genetic algorithms for news filtering. The tf-idf method is also compared against Lang's MDL method as a baseline for evaluation in NewsWeeder (Lang, 1995).

More human-like approaches to news understanding have been explored in symbolic, knowledge-based systems. The advantage of these approaches is that the input text is understood as a human might understand the text, allowing for much greater understanding (Ram, 1992). An early knowledge-based approach to news story understanding is the FRUMP system developed by DeJong (1982). Given UPI news stories, FRUMP processes the story by parsing into a CD representation (Schank and Abelson, 1977), comparing the story with stereotypical events through a structure named a *script*, and provides a summary of the article. A more recent work that also performs script based learning to understand and retrieve usenet news articles is Mauldin's FERRET system (Mauldin, 1991). In Ferret, articles that match the defined scripts may then be disambiguated with the script, classified in terms of their content, and matched with the query. The novel features in Ferret include an online dictionary to augment the understanding process and script learning through genetic algorithms.

Another method of filtering which has recently attracted attention is collaborative or social filtering. This involves the annotation or public review of articles by a population of other users. The reviews then become an input for the filter. As a result, a user may decide to read an article based upon the reaction of his peers; e.g., user A may choose to read articles only examined by user B or user C. Collaborative systems for filtering mail, usenet news, and WWW documents are currently under investigation (Lashkari et. al.,1994; Mock, 1996; Goldberg et. al, 1992).

Global Hill Climbing Filtering Algorithm

The data used for the filtering experiments consisted of 144 sequentially posted news articles from the **ucd.life** newsgroup. This newsgroup was selected since the subjects in the study were 14 UC Davis undergraduates and the newsgroup covers a variety of topics likely to be of interest to the general Davis community. This newsgroup receives approximately 50 messages a day so that filtering may be applicable. When processing articles, the extracted tokens were first passed through a stop list, but not through a stemmer. Additionally, binary encoded files were thrown out, extraneous header information stripped, and quoted material from old articles removed.

Global Hill Climbing - A Keyword Scheme

One of the requirements for the user model is that it must be very simple for users to modify and understand; if the model is too difficult to manipulate, the average user will never use it (Stevens, 1992; Mock, 1996). In addition to simplicity, the model must also provide for good performance. Consequently, a keyword/feature based system was initially selected for the user model since it is easy to perform computationally and also easy for users to understand.

Based upon the requirements of simplicity and user modifiability, a simple classification scheme was implemented in INFOS termed Global Hill Climbing (GHC). This is a linear discriminant method based on a table of features. The table counts the number of times each feature has been found in each class. Since the table contains only one variable per class, it is simple for users to understand and manipulate. The table is created in a hill climbing fashion; as the user reads messages, she indicates whether or not each message read was accepted (liked) or rejected (disliked). The outcome is used to increment the table's weights accordingly.

An example is shown in Table 1. Here, the feature "genetic" has appeared in five accepted articles, the author feature of "grog@ucdavis" has appeared in three accepted articles and one rejected article, etc. This data indicates an interest in articles posted by grog or containing the word "genetic," and a disinterest in articles containing the word "flames." In addition to using tokens from the articles as features, collaborative review features are also included in the table. These other users are local users running the same news system who are willing to share their own reviews with others. In Table 1, the user "Kiki" has accepted four articles the current reader has accepted, and Kiki has rejected one article the current user has accepted. Similarly, Kiki has rejected two articles the current user has accepted, and rejected three articles the current user has rejected. This indicates that the current reader's accepted messages strongly correspond with Kiki's accepted messages. However, the user's rejected messages only slightly correspond with Kiki's rejected messages. The table continues to grow as new articles are read.

Word	Accepted	Rejected
genetic	5	0
algorithm	3	3
flames	2	7
grog@ucdavis	3	1
Kiki Accepted	4	1
Kiki Rejected	2	3

Table 1: Global Hill Climbing Table of Weights

Given such a table, classification of new messages is performed by extracting the features from the new article and then computing the sum of all the Accepted and Rejected values from matching features in the table. If the Accepted percentage minus the Rejected percentage exceeds A, the message is classified as being of interest. Conversely, if the Rejected percentage less the Accepted Percentage exceeds A, the message is classified as being of no interest. Messages in between are marked unknown. In INFOS, A was set to 0.15. However, this value has been left as a user-adjustable setting to allow more aggressive or conservative classifications. Mathematically, the classification process for a set of feature terms *t* is referenced by:

$$SimilarityPercentage(class)_t = \frac{\sum_t ClassOccurrences_t}{\sum_t TotalOccurrences_t}$$

$$Class_t = \begin{Bmatrix} \left(SimilarityPerc(Acc)_t - SimilarityPerc(\mathrm{Re}\,j)_t\right) > A\!: Accepted \\ \left(SimilarityPerc(\mathrm{Re}\,j)_t - SimilarityPerc(Acc)_t\right) > A\!: \mathrm{Re}\,jected \\ else\!: Unknown \end{Bmatrix}$$

The system is similar to the tf-idf method, but it does not explicitly reference the inverse document frequency term as a simplification. However, the inverse document concept is implicitly referenced in the stop list and the accepted/rejected counters.

Assigning Weights

As the algorithm stands, all features are treated equally. Authors, body text, subject text, and collaborative data are all combined identically. While this allows each feature to account for as large or small a contribution as desired, the result is a bias toward those features that occur most often. For example, the word "computer" is more likely to occur in the body of articles in a computer newsgroup, than the author of a particular group. The computer term may appear thousands of times, while an individual author will probably only appear a handful of times. As a result, the contribution from author's terms will be negligible when compared against other more frequently occurring features.

One solution to this problem is to separate the GHC table into a set of individual tables - one table for each type of feature. Percentages of acceptance and rejection can be computed from the features among each table, and then these percentages combined to compute the final classification:

$$SimilarityCombn(Class)_t = \begin{matrix} K_1 \times SimilarityPerc(Class)_{author} + \\ K_2 \times SimilarityPerc(Class)_{sub} + \\ K_3 \times SimilarityPerc(Class)_{text} + \\ K_4 \times SimilarityPerc(Class)_{collaborative} \end{matrix}$$

$$Class_t = \begin{Bmatrix} \left(SimilarityCombn(Acc)_t - SimilarityCombn(\mathrm{Re}\,j)_t\right) > A\!: Accepted \\ \left(SimilarityCombn(\mathrm{Re}\,j)_t - SimilarityCombn(Acc)_t\right) > A\!: \mathrm{Re}\,jected \\ else\!: Unknown \end{Bmatrix}$$

What values should be assigned to constants K_1 through K_4? Some systems (Jennings & Higuchi, 1992) give higher weight to the subject features on the assumption that these are most predictive. To investigate which terms are actually most predictive, experiments were performed to evaluate the impact of each feature individually. The features were then combined based upon how much impact they showed individually; i.e., the most predictive feature was given the highest weights.

To test the feature's contribution to the classifications, users read 100 sequentially posted messages from the ucd.life newsgroup and marked each as accepted, rejected, or unknown. From these 100 messages, 50 messages were randomly selected for training, and the system predicted the users' choices for the rest of the messages among one set of features. These predictions were one of three classes: Suggested, Not Suggested, or Unknown. The predictions were then compared to the actual classifications provided by the subjects. The evaluation metric used in this experiment is classification accuracy: the percentage of predicted articles that were classified correctly.

The experimental results are shown in Table 2. The subject features results in the highest percentage correct with the lowest error, probably since subject words are accurate predictors of entire threads that may be of interest. All schemes perform better than chance or by always predicting the most likely class.

Feature Used Alone	% Correct	% Unknown	% Incorrect
Author	38.4	46.7	14.9
Subject	52.1	35.5	11.8
Textbody	53.6	27.2	19.2
Collaborative	46.2	41.2	12.6

Table 2: Classification accuracy for individual sets of features.

The results from this experiment indicate that the subject features should have the highest weighting, followed by textbody and collaborative data. Author features should have the lowest weighting. A value of 0.35 was assigned to K_2, the subject's weight, 0.25 to K_3 and K_4, the collaborative and textbody weights, and 0.15 to K_1, the author's weight. Using these weights, the classification process was rerun and the results shown below. The error decreased significantly while recall remained constant.

Percentage Correct Classifications	51.5%
Percentage Incorrect Classifications	7.3%
Percentage Unknown Classifications	40.9%
Within Error, Percent of False Positives:	50%
Within Error, Percent of False Negatives:	50%

Case-Based Reasoning Method

The GHC method's main strength lies in its simplicity, user modifiability, and predictive abilities. However, GHC considers words to be conditionally independent from other words. This is certainly not the case for words with multiple meanings.

The method used in INFOS to address these problems is a case-based reasoning system that incorporates semantic knowledge. By retrieving individual cases and using the classification of those cases to classify new articles, the system is capable of avoiding the limitations of linearity. Furthermore, by designing a case-based reasoning system with semantic knowledge, INFOS is capable of comparing concepts rather than individual words. Finally, a CBR system also provides a means for information retrieval in addition to information filtering.

Index Extraction

This work employs both controlled and uncontrolled index extraction as in the CLARIT system (Evans et. al., 1991). In the controlled approach, a predefined list of knowledge structures is used to guide the indexing process. While accurate, this method requires a fully defined knowledge base for all the structures that may occur. Currently, this is not possible for new domains. The uncontrolled approach relies on general purpose methods rather than pre-existing domain knowledge. As a result, indices may not be as well-defined as the controlled approach, but the benefit is generality across domains. INFOS uses a combination of both approaches in an attempt to acquire the benefits of each. The controlled approach in INFOS is composed of a knowledge-based method derived from WordNet, while the uncontrolled approach is composed of a keyword-based inverted index using features such as unknown words, author names, or collaborative data.

INFOS uses WordNet (Miller, 1995) to map words into concepts, and these concepts are used as indices rather than the actual words. In the event that a word is missing from the WordNet lexicon, then that word is used in an inverted index to index the source document directly. To narrow the amount of data required for processing articles, INFOS only focuses upon the verbs and nouns indexed in WordNet.

WordNet is a knowledge-base of English words that includes part of speech identification, synonyms, frequency usage, etc. Concepts are defined in terms of a hierarchical semantic organization; e.g., the word "oak" is defined as a oak-->tree-->plant-->organism, where arrows indicate ISA relationships. WordNet v1.5 contains approximately 107,000 noun senses and 27,000 verb senses - the size of a paperback dictionary. An example of the WordNet ISA hierarchy for the word "ocean" is shown in figure 1. Words are defined in terms of *senses*. In the case of ocean, there are two noun senses; one for the body of water, and the other for a large quantity.

```
SENSE 1
main, ocean, sea, briny
      => body of water, water
          => object, inanimate object, physical object
              => entity
SENSE 2
ocean, sea
      => large indefinite quantity
          => indefinite quantity
              => measure, quantity, amount, quantum
                  => abstraction
```

Figure 1 : Example WordNet hypernym hierarchies for the word "ocean." This word has two sense definitions.

If INFOS indexed news articles based upon all the sense definitions of nouns and verbs found in an article, theı a large number of irrelevant indices would be created due to multiple word meanings. Consequently, INFOS attempts to find appropriate noun or verb phrases based upon Paice's index extraction algorithm (Paice, 1989). filtering. This algorithm assumes that sentences repeat an underlying concept within a "topic neighborhood" of a few sentences. Those words occurring with a high frequency are likely to be relevant to the topic at hand.

Paice's algorithm was modified to operate upon WordNet word sense definitions rather than individual words. First, verbs and nouns from each sentence are identified through WordNet and their sense definition referenced. This step results in a linked list of senses. Since each word is expanded into all possible sense definitions of that word, this pool of sense definitions may not accurately reflect the actual topic. To select relevant senses, a sentence neighborhood is examined and the intersection of sense definitions that match within a specified neighborhood are selected. This process restricts the selected definitions only to those that are reoccurring and are then more likely to be relevant to the document. Only the first 20 sentences of articles were processed to speed execution in the event of extremely long postings. Algorithmic details may be found in Paice's work (1989).

After candidate nouns and verbs have been identified, other relevancy statistics are also associated to each sense term, including frequency and rarity (Evans et. al, 1991) determined through Wordnet and document statistics. Once both frequency and rarity have been determined, the two are multiplied together to give a general relevancy statistic for a sense term. The relevancy value is stored with each term and is used in memory retrieval to determine how closely an old article matches a new document.

Indexing Cases

Once the appropriate phrase senses have been extracted from a textual case, the article is saved and the senses used to index the case. The method in which articles are indexed is to construct a pointer to the file that contains currently defined sense in a global abstraction hierarchy.

An example memory hierarchy with three cases is shown in figure 2. In this example, one article contains the word "vehicle," another article contains the word "bicycle," and the last article contains the word "car." In figure 2, the root node is not shown, but the sub-hierarchy starting at the Conveyance concept is displayed. This node represents the concept regarding items of transport and conveyance. All sub-nodes refine a particular concept and inherit the norms of their ancestors; hence all nodes located below Conveyance must also refer to transportation vehicles. In addition to pointing to sub-nodes, the Vehicle node also has an index to a specific case (news article) referencing vehicles. In a similar fashion, indices from the wheeled vehicle and the auto nodes are further specialized until they also point to actual cases .

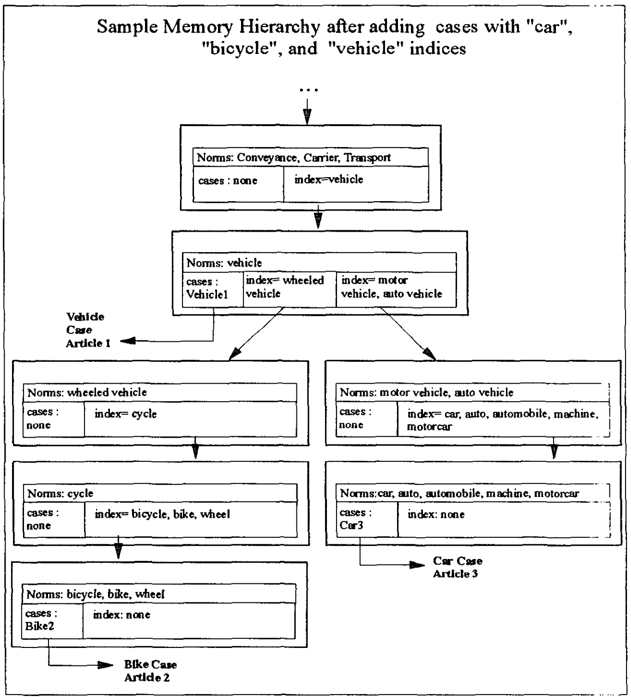
Sample Memory Hierarchy after adding cases with "car", "bicycle", and "vehicle" indices
...
Norms: Conveyance, Carrier, Transport
cases : none
index=vehicle
Norms: vehicle
cases : Vehicle1
index= wheeled vehicle
index= motor vehicle, auto vehicle
Vehicle Case Article 1
Norms: wheeled vehicle
cases : none
index= cycle
Norms: motor vehicle, auto vehicle
cases : none
index= car, auto, automobile, machine, motorcar
Norms: cycle
cases : none
index= bicycle, bike, wheel
Norms:car, auto, automobile, machine, motorcar
cases : Car3
index: none
Car Case Article 3
Norms: bicycle, bike, wheel
cases : Bike2
index: none
Bike Case Article 2

Figure 2: Sample Memory Hierarchy for Indexing Cases

Memory Retrieval

Case-based memory retrieval involves searching for applicable cases based upon a given set of features. These features are simply WordNet sense indices from a new article that needs to be classified. Case retrieval in INFOS simply involves a depth-first search in the memory hierarchy along indices that match the input query. To allow for partial matches (e.g., retrieve cases regarding bicycles when the input is about cars), path mismatches are traversed until an error threshold value is exceeded.

For each retrieved case, the match value is computed by summing over all n feature queries the distance function:

$$Match = \frac{1}{n}\sum_{i=1}^{n}(MatchPercent_i) \times \mathrm{Re}\,levancy_i$$

In INFOS, the retrieved cases are sorted by degree of match. The classification statistics of the best matching case can then be used to classify the new article, using the Accepted and Rejected counters for the case and computing as described in the Hill Climbing section. The case article can also be displayed as a justification to the proposed classification of the new article.

Results of Case-Based Scheme

The same testing methodology that was used to evaluate the GHC scheme was also run with the case-based scheme. Finally, the case-based scheme was tested when used in conjunction with the global scheme. In this mode, the global scheme classification was performed first. If the global scheme returned an unknown classification, then the classification of the case-based scheme was used. The global scheme was performed first since it is quicker to evaluate than the CBR method and still retains a low error rate.

A summary of the results is shown in Table 3 depicting the GHC method, CBR alone using the best matching case, and CBR combined with the GHC method.

Method	% Corr.	% Unk.	% Incorr.	% FP	% FN
GHC	51.5	40.9	7.3	50	50
CBR	39.8	50.5	9.5	77	33
GHC + CBR	58.0	29.9	12.1	62	37

Table 3: Classification accuracy for GHC, CBR, and Hybrid Methods. Values displayed are % Classified Correctly, Unknown, Incorrectly, and within the error, % of False Positives and False Negatives.

The results from this experiment indicate that the GHC method still has the lowest error but the combined scheme provides the best correct classification rate. The CBR scheme will have some poor indices due to the sense disambiguation problem that can allow irrelevant cases to be retrieved. Consequently, the CBR method has a higher error rate than the global hill climbing method. When combined with the global hill climbing scheme, the best match CBR method does achieve a higher correct classification percentage at 58%, although it suffers from a slightly higher error rate of 12%. These mixed results show potential for the hybrid method, but indicates the need for more robust word sense disambiguation.

Future and Ongoing Work

Ongoing work with INFOS is incorporating genetic algorithms to explore the news space and index patterns to disambiguate input text more accurately. Other areas of proposed work include modifications for INFOS to run offline, a graphical user interface, self-modifying parameters, new knowledge bases, and the application toward the WWW and intelligent tutoring systems. More information is available from http://phobos.cs.ucdavis.edu:8001/~mock/INFOS/infos.hml.

References

DeJong, G. 1982. An Overview of the FRUMP System. In Lehnert, W., and Ringle, M. eds. *Strategies for Natural Language Processing*, Hillsdale, NJ: Lawrence Erlbaum.

Evans, D.A.; Ginther-Webster, K.; Hart, M. Lefferts; and Monarch, I.A. 1991. Automatic Indexing Using Selective NLP and First-Order Thesauri. In Proceedings of the Intelligent Text and Image Handling Conference, 624-643, Barcelona, Spain.

Goldberg, D.; Nichols, D.; Oki, B. and Terry, D. 1992. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM* 35:12,61-70.

Jennings, A. and Higuchi, H. 1992. A Personal News Service Based on a User Model Neural Network. *IEICE Transactions Inf. & Systems* E75:D(2), 198-209.

Lang, K. 1995. NewsWeeder: Learning to Filter Netnews. In Proceedings of the Twelfth International Machine Learning Conference.

Lashkari, Y.; Metral, M., and Maes, P. 1994. Collaborative Interface Agents. In Proceedings of the Twelfth National Conference on Artificial Intelligence, 444-449.

Mauldin, M. L. 1991. *Conceptual Information Retrieval: A case study in Adaptive Partial Parsing*. Norwell, MA: Kluwer Academic Publishers.

Miller, G.A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 38:11, 39-41.

Mock, K.J. 1996. Intelligent Information Filtering via Hybrid Techniques: Hill Climbing, Case-Based Reasoning, and Index Patterns. Ph.D. Dissertation, Dept. of Computer Science, University of California, Davis.

Paice, C.D. 1989. Automatic Generation and Evaluation of Back-of Book Indexes. *Prospects for Intelligent Retrieval, Informatics 10.*

Ram, A. 1992. Natural Language Understanding for Information-Filtering Systems. *Communications of the ACM* 35:12, 80-81.

Salton, G. 1971. *SMART Retrieval System: Experiments in automatic document processing*. Englewood Cliffs, NJ: Prentice-Hall.

Schank, R.C. and Abelson, R. 1977. *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Lawrence Erlbaum .

Sheth, B.D. 1994. A Learning Approach to Personalized Information Filtering. Masters Thesis. Dept of Computer Science and Engineering, Mass. Institute of Technology.

Stevens, C. 1992. Knowledge-Based Assistance for Accessing Large, Poorly Structured Information Spaces. Ph.D. Dissertation. Dept of Computer Science, University of Colorado.