Finding Salient Features for Personal Web Page Categories

Marilyn R. Wulfekuhler and William F. Punch

Genetic **A**lgorithms **R**esearch and **A**pplications **G**roup, the **GARAGe**
A714 Wells Hall, Michigan State University, E. Lansing MI 48824
wulfekuh@cps.msu.edu, punch@cps.msu.edu
`http://isl.cps.msu.edu/GA`

Abstract

We examine techniques that "discover" features in sets of pre–categorized documents, such that similar documents can be found on the World Wide Web. First, we examine techniques which will classify training examples with high accuracy, then explain why this is not necessarily useful. We then describe a method for extracting word clusters from the raw document features. Results show that the clustering technique is successful in discovering word grouops which can be used to find similar information on the World Wide Web.

1 Introduction

The explosive growth of the Internet and the World Wide Web has changed the way we think about and seek information. The web provides an enormous resource for every imaginable topic, but does not provide a good means to find the information relevant to one's own interests. There are characteristics of the World Wide Web that make it extremely powerful, but also present significant challenges to users looking for particular information. First, information sources are distributed all over the world, and are not organized according to any agreed upon indexing scheme. Secondly, the information is heterogeneous and can take many different forms. Thirdly, the Web is dynamic, with the potential for both content and location to change at any time.

Current web searching is based on traditional information retrieval techniques and is typically based on boolean operations of keywords. Major web search services such as Lycos [1], Alta Vista [2], WebCrawler [3], and Open Text [4] employ software robots (also called spiders) which traverse the web and create an index based on the full text of the documents they find. A user submits a keyword query to one of these services, and receives the location of any document found in the index which matches the query. Different services vary in their methods of constructing the index, which is why identical queries to different search engines produce different results. Query format is also different for various services. If a user is skilled enough to formulate an appropriate query, most search engines will retrieve pages with adequate *recall* (the percent of the relevant pages retrieved among all possible relevant pages), but with poor *precision* (the ratio of relevant pages to the total number of pages retrieved).

Most users are not experts in information retrieval and can not formulate queries which narrow the search to the context they have in mind. Furthermore, typical users are not aware of the way the search engine designers have constructed their document indexes. Indexes are constructed to be general and applicable to all; they are not tailored to an individual's preferences and needs. They necessarily include all senses of a key search term's usage, which often results in irrelevant information being presented. We want to be able to discover salient features of already established document categories as defined in a user's personal bookmark file. Once these features have been discovered, we can create a software agent which can use existing web search tools on behalf of an individual user to automatically seek new web documents which are potentially interesting, while filtering out much of the unnecessary information.

2 Problem Description

Web documents are grouped together into categories (in the form of a user's bookmark file) according to an individual's preferences; there is no absolute ground truth of classification except what is defined by a particular user. Two people may have the same collection of documents, but classify them differently with both groupings equally valid. The criteria that an individual uses to classify documents are related to some underlying semantic concepts which are not directly measurable as features of the documents. However, the *words* that compose the documents are measurable, and we use them as the *raw features* of the documents. Our goal is to automatically discover some higher level features, some function of the raw features, that represent the underlying semantic concepts which led to the user's particular classification. Clues that will help lead to discovery of the semantic concepts are in the user's classification of the bookmarked documents, so one test of feature discovery is to try to find features/words such that classification of the training documents matches the user's original categorization.

If we find features that reproduce the user's classification of bookmarks, we should be able to use those features to obtain additional documents which are similar and fit into the classification. However, we have found that is not necessarily the case. We can find features among the raw features which reproduce the given classification with high accuracy, but are not useful in capturing semantic concepts and therefore not useful in retrieving similar documents, so focusing on classification accuracy alone is not sufficient.

We have also examined techniques other than just classification of documents which *do* discover features useful for document search which we will describe in Section 3.3.

2.1 Pattern Recognition and Clustering

Pattern recognition is a mature field in computer science with well established techniques for the assignment of unknown patterns to categories, or classes. A pattern is defined as a vector of some number of measurements, called features. Usually a pattern recognition system uses training samples from known categories to form a decision rule for unknown patterns. The unknown pattern is assigned to one of the categories according to the decision rule. Since we are interested in the classes of documents that have been assigned by the user, we can use pattern recognition techniques to try to classify previously unseen documents into the user's categories. While pattern recognition techniques require that the number and labels of categories are known, *clustering* techniques are unsupervised, requiring no external knowledge of categories. Clustering methods simply try to group similar patterns into clusters whose members are more similar to each other (according to some distance measure) than to members of other clusters. There is no a priori knowledge of patterns that belong to certain groups, or even how many groups are appropriate. Refer to basic pattern recognition and clustering texts such as [5, 6, 7] for further information.

We first employ pattern recognition techniques on documents to attempt to find features for classification, then focus on clustering the raw features of the documents.

2.2 Sample Data

Our sample problem comes from the manufacturing domain, with web documents from the Network for Excellence in Manufacturing (NEM Online)[1]. The sample data set from NEM Online consists of 85 web documents in 4 categories, as shown in Table 1.

The category labels were assigned by human experts. Note that the categories may contain underlying concepts which overlap. For example, a document discussing affirmative action may be reasonably classified into government, labor, or legal categories, according to the person who is doing the category assignment, and within the context of their other collected documents.

In pattern recognition terminology, each web document is a *pattern*, stems of words are the *features*, and the number of occurrences of words in the document form the *feature values*. The 85 documents contain a total of 5190 distinct word stems (defined in the next section), so there are 85 patterns in 5190 dimensional feature space. A portion of the 85×5190 pattern matrix is given in Table 2.

To classify the documents using traditional pattern recognition techniques, we use some distance measure to group the pattern vectors which are "close" to each other in the feature space.

[1] http://web.miep.org:80/miep/index.html

Category	Number of Documents
design	23
government	15
labor	13
legal	34
total	85

Table 1: Sample data set categories

The conventional rule of thumb for statistical pattern recognition techniques is that you need five to ten times as many training samples as features for each class in order to estimate the class probability distributions [8]. For our 4 class problem and 5190 features, this means we would need one to two hundred thousand training documents. Clearly this is not possible for this domain.

85 points in 5190 dimensional space is extremely sparse, and an individual pattern vector will contain a value of 0 in many of the dimensions, since not all words in the global space will appear in every document. We know that the 5190 dimensions are not uncorrelated, and not independent. In order to analyze this data, and find a function of the raw features which will be semantically useful, we need to reduce the dimensionality of the problem somehow. Reducing the dimensionality of the problem will make classification manageable, and will also aid in finding the salient features which will be most useful for subsequent search of new documents.

We discuss in Section 3.2 how we reduce dimensionality through standard feature selection techniques, and in Section 3.3 we discuss clustering the existing features into smaller groups.

3 Pattern Analysis of Web Documents

We first explore feature selection, where a "good" subset of the raw features is selected in order to reduce the dimensionality and also improve classification performance. However, improving classification performance by itself is not enough unless we can use the features discovered to retrieve additional documents that correspond to the user's categories.

We then describe a clustering technique which we use to group features (which have no prior established categories) rather than documents. This technique worked quite well for discovering features which represent finer grained concepts than the broad categories of labor, legal, government, and design.

3.1 Preprocessing

First we parse a bookmark list to obtain the category assignments and the URLs of each document. We retrieve and store the documents locally; these form our document *training set*.

We then parse each document for *words*, defined as a contiguous string of letters, ignoring HTML tags, digits, and punctuation, as well as common English words (such as *the, of, and*, etc) from a pre–defined stop list. Our stop list is the one given by Fox [9]. We then reduce each remaining word to its word stem, so that words like *computer* and *computing* both become the same term, *comput*. This is a standard information retrieval technique, and we use the algorithm from Frakes[10]. All unique word stems from the entire training set of documents form the global feature space.

Stemming the words has the obvious advantage of reducing the feature space, resulting in 5190 distinct stemmed features vs. 7633 distinct non–stemmed features in our 85 documents. It would also seem advantageous in classification, since words with the same stems should be related to the same semantic concept, so we should count them as the same feature without regarding slight variations in tense and usage as distinct features. However, there are also disadvantages to word stemming. There are cases where the simple stemming algorithm reduces words to the same stem when the originals were not related. For example, "animal" and "animation" are both reduced to the same stem, "anim". There are few contexts where they refer to the same concept throughout a document collection, so counting them as the same feature may negatively

Category	Doc Num	action	affirm	discrimin	drug	fda	banana	central	cereal	chairman
	25	7	0	0	0	0	0	0	0	0
	27	7	0	0	42	33	0	0	0	0
legal	28	53	0	0	2	5	1	0	5	0
	29	3	0	0	0	47	0	0	0	0
	30	1	0	0	5	27	0	1	0	0
	41	2	0	1	0	0	0	0	0	0
	42	1	0	10	0	0	0	0	0	0
	44	1	0	0	0	0	0	0	0	0
	45	2	0	0	0	0	0	0	0	0
labor	46	31	31	15	0	0	0	0	0	0
	47	52	50	12	0	0	0	0	0	0
	48	4	4	0	0	0	0	0	0	0
	49	2	2	0	0	0	0	0	0	0
	50	51	49	23	1	0	0	2	0	0
	53	5	0	0	0	0	0	0	0	0
	58	4	1	0	0	0	0	0	0	0
	59	5	0	0	0	0	0	2	0	2
	60	2	0	0	0	0	0	0	0	2
government	66	0	0	0	1	0	0	0	0	0
	68	2	0	0	0	0	0	0	0	0
	70	0	0	0	3	0	0	0	0	1
	71	2	0	0	4	1	0	0	0	0
	72	14	0	0	0	0	0	0	0	0
	76	0	0	0	0	0	0	1	0	0

Table 2: Sample of the partial pattern matrix. Rows represent patterns (documents) and columns represent features (word stems). Selected features out of the 5190 total feature space are shown for illustrative purposes. Rows which are not shown have 0 values in all positions of those columns.

impact classification results. The other disadvantage to stemming is the loss of the original term, which can not then be used in subsequent web searching. For example, in the word "feed" the "ed" is removed to get a stem of "fe". Doing a web search on the string "fe" will produce many more documents than are relevant to "feed". Despite the potential pitfalls, we used stemming in order to reduce the initial feature space as much as possible, but will examine the stemming issue more in the future.

3.2 Classification of Documents

The first most obvious step in attempting to classify patterns in such a high dimensional space is to simply select a "good" subset of the raw features. There are 2^n possible subsets of n original features, so finding the optimal subset exhaustively is computationally prohibitive, especially when $n = 5190$.

Recall that in our pattern matrix shown in Table 2, rows represent documents, columns represent individual features (word stems), and the feature values are the number of occurrences of that word in the given document. When trying to categorize the documents, we find those row vectors which are similar to each other and assign an unknown document a label according to the row that is most similar.

If we refer to the pattern matrix in Table 2, we can see that rows 25 through 30, which are all from the "legal" category, are similar (and are thus close to each other in the feature space). We can easily see this because we are looking at a vastly truncated pattern matrix, with rows and columns selected for illustrative purposes. The problem with classifying the row vectors comes when we include all 5190 features, which makes it much more difficult to identify similar rows.

For our pattern classifier, we used Euclidean distance as the distance measure between documents, and a nearest neighbor decision rule. When presented with an unknown pattern, we examine the training patterns,

and determine the one which is the closest in Euclidean distance. The unknown pattern is then assigned to the category of this *nearest neighbor*. We used leave one out testing to evaluate the performance of the classifier on various feature sets.

The above classifier using all 5190 features misclassified 50 out of 85 documents, for an accuracy of 41.18%. We found we can improve the classification accuracy through feature selection techniques. Some effective conventional methods for feature selection are sequential forward selection [5, 11], sequential floating feature selection [12], and genetic algorithm search [13, 14].

Sequential forward selection achieved 17/85 errors, or 80% accuracy by selecting 13 features. These features were *engin, action, david, contempl, affirm, architectur, ave, osha, abund, rehabilit, notic, commerc, transact.*

The genetic algorithm feature selection method used a fixed size subset of 25, and the best such subset achieved 21/85 errors for 75.29% accuracy. The feature set was *bliznakov, volant, law, biographi, permiss, cwatx, alaska, treatment, sodium, move, evolut, version, avex, darnell, thoma, photographi, build, pany, hospit, law, lexi, plamen, briefli, export, cite* .

Selecting any number of features sequentially, we were able to ultimately achieve 7/85 errors, for a 91.76% accuracy rate with only 17 features. Those features are *project, job, staff, david, apr, clinton, govern, gov, sophist, affirm, favor, commission, eeo, fiscal, cfo, tm, stori.*

Though the subsets of features found through all three feature selection techniques achieved accurate classification of the training data, the features in each subset were quite different and do not by themselves suggest the categories of "labor", "legal", "design", and "government" in the manufacturing domain.

It is not meaningful to select a feature as a category discriminator when there are many potential candidate neighbors for a test pattern with few features. Why then do the selected sets of features above achieve such high accuracy rates? The answer lies in the sparse nature of the original pattern space. With 85 points in 5190 dimensional space, there will be many piecewise linear decision boundaries which adequately partition the training patterns. However, the real question is whether these decision boundaries are meaningful or generalizable. Recall that accurate classification of the existing documents is not sufficient in our ultimate goal. We want not only to find some decision boundary, but the best general decision boundary, and meaningful features which will enable us to find new documents.

3.3 Feature Clustering

Our efforts at classification accuracy using feature selection were successful, but the resulting features were not useful in seeking new documents or in suggesting semantic concepts. Simple selection of raw features is not powerful enough to capture context or concepts. Since we know that the features in the original pattern matrix are not uncorrelated, we should be able to cluster the 5190 *features* into groups whose members are similar to each other and not similar to members of other groups.

Where previously we were trying to assign the *documents* to a group according to the category of its neighbors in the word stem space, we will now try to find groups of *word stems* which occur in the document space. Since there are no existing categories of words stems, grouping the features is necessarily done *without* regard to any category labels. Now instead of looking for similarities between row vectors in the pattern matrix (Table 2), we look for similarities between column vectors. We can think of the word stems as the patterns, and the documents they occur in as the features. Instead of grouping 85 patterns in 5190 dimensional space, we can imagine grouping 5190 points in 85 dimensional space, which is much less sparse.

Referring to Table 2, we see that the columns for the features "action", "affirm", and "discrim" are similar. In this case, documents 46, 47, and 50 contained similar values for those terms. Even though the algorithm has no knowledge of document categories, there are some terms which have similar occurrences throughout the document set. This causes the column vectors to be similar (near each other in the *document* space) and thus the terms are put into the same group.

To group the features, we used Hartigan's K–means partitional clustering algorithm [15] as implemented by S–PLUS[2], where K points are chosen randomly as the means of K groups in the 85 dimensional space. Each of the 5190 points is then assigned to the group whose mean is closest in Euclidean distance. After each point is assigned, the group means are recomputed, and the process repeats until either there is no change, or after a fixed number of iterations. The K–means process is not guaranteed to find a global optimum

[2]S–PLUS is a registered trademark of Statisitcal Sciences, Inc.

grouping, so we run it several times with different random seeds in order to increase our confidence in the resulting clusters.

We chose a value of $K = 2(c+1)$, where c is the number of original document categories. With a value of $K = 2(4+1) = 10$, we assign each of the 5190 features to one of 10 groups. We ran 10 times with different random seeds, and results were consistent. One of the surprising results was that there was always a single group which contained an average of 92% of the words. These are the features which are useless for discriminating among the documents. If we discard this largest group, we are left with around 400 features to consider. The sizes of all 10 clusters for a typical run were 3, 6, 9, 9, 10, 11, 13, 104, 243, 4791. The smallest clusters are particularly interesting in that they contain word stems which to a human seem to be semantically related. The 7 smallest cluster contents in the above clustering are shown in Table 3.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
employe	applic	action	cadmaz	cfr	amend	anim
fmla	claim	affirm	consult	contain	bankruptci	commod
leav	file	american	copyright	cosmet	code	cpg
	invent	discrimin	custom	ey	court	except
	patent	job	design	hair	creditor	fat
	provision	minor	manag	ingredi	debtor	fe
		opportun	project	label	petition	food
		peopl	sect	manufactur	properti	fruit
		women	servic	product	section	level
				regul	secur	ppm
					truste	refer
						top
						veget

Table 3: The seven smallest clusters found in the document set. These are stemmed words.

3.3.1 Discussion

One result that is interesting is that the two terms "affirm" and "action" nearly always appeared in the same cluster (in nine out of ten runs). If we look at the original pattern matrix (Table 2), we see that the term "action" appears in 22 different documents, and a total of 255 times. The term "affirm" appears in only 6 documents, a total of 137 times. However, in those 6 documents, "action" occurs almost exactly the same number of times as "affirm", which caused the clustering algorithm to consider the two column vectors similar, and group the two terms together, even though "action" by itself occurred in many other documents. We know that "affirmative action" is a two word phrase, and it appears that the clustering algorithm successfully grouped the terms together.

Also, the term "file" appears in a cluster 2. "file" by itself has many meanings, including a computer file. However, when we see the *context* of the remaining words in the cluster, it is apparent that this sense of "file" refers to the filing of patent application.

Terms which are used in ordinary contexts, or unique terms which don't occur often across the training document set, will tend to cluster into the large 4000 member group. This takes care of spelling errors, proper names which are infrequent, and terms which are used in the same manner throughout the entire document set. Terms used in specific contexts (such as file) will appear in the documents consistently with other terms appropriate to that context (patent, invent) and thus will tend to cluster together. Among the groups of words, unique contexts stand out from the crowd.

We now have a smaller set of features which can be used to construct queries for seeking out other relevant documents. If we use the conjunction of terms from a single cluster as a query to existing Web search engines, we get many documents which contain those terms, in the appropriate context. Searching the web with word clusters allows discovery of finer grained topics (eg, family medical leave) within the broadly defined categories (eg, labor).

Cluster 1	Cluster 2	Cluster 3		Cluster 4	
iso	patent	busi	commerci	corpor	design
		export	financ	engin	home
		govern	help	inform	intern
		onlin	program	office	page
		resourc		servic	system
				utsi	
Cluster 5		Cluster 6		Cluster 7	
com	congress	antitrust	comparabas	action	affirm
courtesi	email	comparison	copyright	bill	budget
furnish	html	directori	dot	cfr	code
http	jhoffman	econom	inc	experiment	feder
move	tm	nec	new	fiscal	homepag
turnpik	url	polici	product	hous	internet
volant	webslingerz	research	sect	law	legisl
webspac	www	site	standard	librari	link
		topic	usa	regul	relat
		web	welcom	search	server
				unit	use

Table 4: The seven smallest clusters based on a row normalized pattern matrix.

Currently evaluation of the results of queries from the word clusters must be done by a human, but we hope to incorporate automatic relevancy ranking of pages to enable the user to more effectively browse for the documents which may be of interest. We are also investigating functions of cluster terms other than conjunction in constructing queries. More importantly, we are also working on ways to relate the word clusters semantically so that we can find relevant documents that contain none of the cluster words, but have the same "meaning".

3.3.2 Normalization

One might wonder why we used number of occurrences of words in the document to form the feature values, rather than the frequency of a word (number of occurrences of a word divided by the total number of words in a given document). It is natural to think of normalizing the documents by dividing each entry of the pattern matrix by the row total, in order to normalize for document length. Under the current scheme, longer documents weigh more heavily into the clustering than shorter ones.

The K–means clustering of words was performed on the same data which had been row normalized. The semantic content was less striking than when using word occurrences. Clustering with a row normalized pattern matrix and the same random seed as the run reported above yielded clusters of sizes 1, 1, 9, 11, 16, 20, 22, 51, 116, 4943. The contents of the 7 smallest clusters are shown in Table 4.

Note that the two words "affirm" and "action" still show up in the same cluster, but related words which were in the same cluster using the unnormalized pattern matrix, such as "job", "women", "minority", and "discrimin", are missing. Many other terms which are probably unrelated to the topic/concept are included, such as "homepag" and "internet".

Normalization seems like a good idea but upon reflection, we realize that it is beneficial only if the documents are fairly homogeneous in their style. We know that HTML documents vary widely, from full text to a simple list of links. We conjecture that we need some length to the documents in order to extract context which will lead to grouping of words from the same concept. Long documents are more likely to reflect "typical" usage patterns than simple lists or abbreviated documents such as abstracts. Long documents are providing more information, so it is reasonable that they contribute more heavily to the clustering. However, we would still like to ensure that the clustering is not relying on length too heavily, which it may be by using raw word counts in the pattern matrix.

Cluster 1	Cluster 2	Cluster 3					
emplye	action	act	administr	african	agenc	ago	am
fmla	affirm	america	applaus	applic	asid	base	black
leav	american	believ	busi	chanc	chang	child	children
	peopl	civil	class	colleg	common	commun	compani
		countri	court	dai	decision	depart	develop
		disabl	discrimin	divers	don	econom	educ
		employ	equal	evid	exampl	fair	famili
		feder	futur	gender	goal	govern	happen
		help	hispan	histori	individu	issu	job
		law	learn	live	loan	male	mean
		meet	middl	million	minor	move	nation
		opportun	own	percent	person	poor	presid
		program	qualifi	question	quota	race	re
		reduc	requir	revers	review	right	school
		set	simpli	standard	stronger	system	time
		treat	white	women	world	wrong	

Table 5: The three smallest clusters from the labor category.

3.3.3 Clustering One Class at a Time

We also investigated what would be the results of feature clustering if we only considered one document category at a time, rather than all categories simultaneously. This means that the dimensionality of the clustering space will be different for each category. For example, the "labor" category consists of 13 documents, which contain 1839 distinct word stems. So we are clustering the 1839 terms into K groups in 13 dimensional space. We choose $K = 4$ using the same heuristic as with the total collection, ie, $2(c+1)$ groups where c is the number of categories. Using $K > 4$ tends to generate empty clusters. Using word counts for feature values in the "labor" category, we get clusters of size 3, 4, 95, and 1737 with the 3 smallest clusters shown in Table 5.

With frequencies used as feature values (the pattern matrix has been row normalized) in the single category case, we get larger clusters which contain the ones generated from unnormalized data. The cluster sizes for "labor" for a typical run were 23, 36, 48, and 1732. The three smallest clusters are shown in Table 6.

It appears that the effect of long documents skewing the clustering is even more pronounced in the case of single category clustering. It seems that for the single category case, word counts as feature values generate overly specific clusters, and frequencies as feature values generate overly general clusters.

Clustering by single category seems to be a good idea and may help in forming clusters from terms which have multiple meanings. For example, when terms from all four classes were clustered, the term "circuit" often appears in same cluster as "cadmaz", "softwar", and "electron", which seems to be consistent with the "circuit design" meaning. Clustering the terms from the "legal" category alone resulted in a cluster which included "circuit", "court" and "appeal", consistent with the legal meaning of circuit in the phrase "circuit court of appeals". Clustering from single categories provides one mechanism for allowing the term "circuit" to be in a feature cluster which is relevant to both the "design" and "legal" categories.

However, when clustering words from a single document category, we must be careful to provide enough diversity in the training data to be able to distinguish different concepts. If a category is too small or too specialized, all the important terms which would be necessary for discrimination among other categories/contexts may be used in the same manner in the smaller training set, and thus get lost in the largest cluster.

Cluster 1			Cluster 2			Cluster 3		
action	affirm	bill	access	advantag	answer	appear	appendic	appendix
code	congress	empoly	assist	ccg	complet	cfr	civil	compens
equal	homepag	hous	connect	elcom	employe	content	contract	control
inform	internet	law	establish	fmla	headquart	directli	director	docum
legisl	link	rd	help	hr	hrhq	dol	draft	ergonom
relat	repress	search	instantli	leav	look	ergoweb	exposur	factor
senat	th	thoma	onlin	password	pleas	feder	health	index
track	use		preview	proven	provid	job	keyword	labor
			reserv	resourc	return	manag	medic	mine
			right	select	servic	oasam	osha	page
			sign	straight	tap	paragraph	pbgc	pertain
			valuabl	welcom	worldwid	propos	protect	public
						pwba	regul	risk
						safeti	section	standard
						tabl	titl	train
						usdol	vet	workplac

Table 6: "Labor" clusters using row normalized data.

4 Conclusions

Characteristics of the World Wide Web and the need for personal organization present challenges for traditional information retrieval and pattern recognition techniques for organizing and finding data. We have shown two important results. First, the standard approach of classifying training documents can be done with high accuracy, but is not generalizable for classification of documents not in the training set, and is not useful in discovering new data. Secondly, clustering words in a document space is effective because it detects small sets of words used in a consistent nonordinary way, and these context clusters can subsequently be used for searching for new information.

Acknowledgements

We acknowledge the support of the Network for Excellence in Manufacturing Online, through the Technology Reinvestment Project (administered by the National Aeronautics and Space Administration), the State of Michigan, Michigan State University, the Edward Lowe Foundation, the Small Business Development Center, Merit Inc., and Great Lakes/Ameritech.

References

[1] Lycos, http://www.lycos.com.

[2] Alta Vista, http://altavista.digital.com.

[3] WebCrawler, http://www.webcrawler.com.

[4] Open Text, http://index.opentext.net.

[5] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Englewood Cliffs, New Jersey: Prentice–Hall, Inc., 1982.

[6] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* Englewood Cliffs, New Jersey: Prentice Hall, 1988.

[7] M. R. Anderberg, *Cluster Analysis for Applications.* New York: Academic Press, 1973.

[8] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics, Vol. 2* (P. R. Krishnaiah and L. N. Kanal, Eds.), pp. 835–855, Amsterdam: North–Holland Publishing Company, 1982.

[9] C. Fox, "Lexical analysis and stoplists," in *Information Retrieval Data Structures and Algorithms* (W. B. Frakes and R. Baeza-Yates, Eds.), pp. 102–130, Englewood Cliffs, New Jersey: Prentice Hall, 1992.

[10] W. B. Frakes, "Stemming algorithms," in *Information Retrieval Data Structures and Algorithms* (W. B. Frakes and R. Baeza-Yates, Eds.), pp. 131–160, Englewood Cliffs, New Jersey: Prentice Hall, 1992.

[11] A. Whitney, "A direct method of nonparametric measurement selection," *IEEE Transactions on Computers*, vol. 20, pp. 1100–1103, 1971.

[12] F. J. Ferri, P. Pudil, M. Hatef, and J. Kittler, "Comparative study of techniques for large–scale feature selection," in *Pattern Recognition in Practice IV, Mutliple Paradigms, Comparative Studies and Hybrid Systems* (E. S. Gelsema and L. S. Kanal, Eds.), pp. 403–413, Amsterdam: Elsevier, 1994.

[13] W. F. Punch, E. D. Goodman, M. Pei, L. Chia-Shun, P. Hovland, and R. Enbody, "Further research on feature selection and classification using genetic algorithms," in *International Conference on Genetic Algorithms 93*, (Champaign, IL), 1993.

[14] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large scale feature selection," *Pattern Recognition Letters*, vol. 10, pp. 335–347, November 1989.

[15] J. A. Hartigan and M. A. Wong, "A k–means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.