Extracting Spatial Knowledge from the Web

Yasuhiko Morimoto* Masaki Aono Michael E. Houle
IBM Tokyo Research Laboratory
1623-14 Shimo-tsuruma, Yamato
Kanagawa 242-8502, Japan
morimoto@acm.org aono@jp.ibm.com meh@trl.ibm.com

Kevin S. McCurley
IBM Almaden Research Center
650 Harry Road, San Jose
California 95120-6099, USA
mccurley@almaden.ibm.com

Abstract

The content of the world-wide web is pervaded by information of a geographical or spatial nature, particularly such location information as addresses, postal codes, and telephone numbers. We present a system for extracting spatial knowledge from collections of web pages gathered by web-crawling programs. For each page determined to contain location information, we apply geocoding techniques to compute geographic coordinates, such as latitude-longitude pairs. Next, we augment the location information with keyword descriptors extracted from the web page contents. We then apply spatial data mining techniques on the augmented location information to derive spatial knowledge.

1. Introduction

The world-wide web is one of the largest information sources of any kind. The content of the web is pervaded by information of a geographical or spatial nature, particularly such location information as addresses, postal codes, telephone numbers and so forth. Commercial web pages typically contain address and contact telephone numbers as well as descriptions of the products and services offered. The introductions of news articles appearing on web pages often state the locations where the events took place, or where they were reported from. It is natural to assume that associations exist between the general contents of a web page and the specific location information it may contain. This paper addresses the problem of extracting spatial knowledge from web content.

1.1. Geospatial Association

Table 1 shows an example of a *geospatial association* extracted from web pages by means of a web crawling program.

Table 1. Geospatial Association

ID	Coordinates	Concepts
705	(37.260 -121.919)	('diet' ···)
6062	(37.890 -122.259)	('diet' ···)
1858	(37.772 -122.414)	('biotech' ···)
···	···	···

In the geospatial association format, the `ID` is the unique identification number of a web page, the `Coordinates` values are geographic coordinates (latitude-longitude pairs), and the `Concepts` values are lists of representative keywords for the web page. Each record of the table implies the existence of a web page whose contents relate to these concepts as well as including information referring to a location. We present the details of how a geospatial association can be extracted later in this paper.

1.2. Application Examples

A geospatial association is a (typically very large) set of two dimensional point objects augmented with concepts. We have developed spatial data mining functions and spatial optimization functions for such two dimensional point objects for the purpose of extracting spatial knowledge.

Let us consider the example of a business database containing records of product sales for each branch of a company. Usually, the addresses of the branches are stored as text strings. In many countries, including the United States, such address information can easily be mapped into a two-dimensional geographic location that can be used to form a geospatial association. Such business databases can thus serve as good sources of geospatial association.

The following functions can all be applied to a geospatial association to discover spatial knowledge.

Neighboring Class Set

The concepts associated with records of a geospatial association form the basis of a classification of these records. Each element of the `Concepts` list is considered to be a class label of the corresponding point. In our example,

*Currently with Hiroshima University

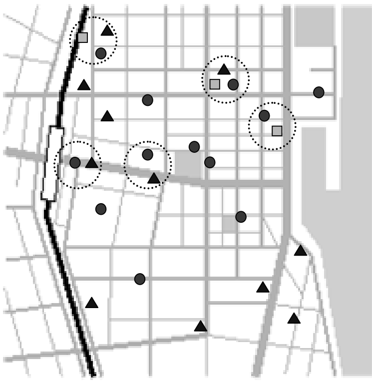


Figure 1. Neighboring Class Set

records of the business database could be classified into "profitable branches" and "unprofitable branches".

The *neighboring class sets* function finds sets of classes whose objects are spatially close to one another [12], according to some minimum distance threshold. Assume that point objects from three classes — **circle**, **triangle** and **square** — are distributed on a map as in Figure 1. In this example, there are four occurrences of a **circle** point situated close to a **triangle** point. Similarly, there are three occurrences of a **circle** point lying near a **square** point, and two occurrences of a **triangle** point appearing next to a **square**. Moreover, there are two occurrences in which all three kinds of points lie close to one another. The set {**circle**, **triangle**} is an example of a neighboring class set whose frequency is four (within the map). The neighboring class sets function enumerates all such sets whose frequency is at least as large as a user specified threshold value, often called the *minimum support*.

The function may find a frequent neighboring class set, for example,

```
({"profitable branch",
"education", "sports"}, 250),
```

which indicates there are 250 instances (triples) consist of a "profitable branch" object, an "education" object, and a "sports" object. From the high frequency of the neighboring class set, we may deduce a relationship among education, sports, and the profitability of the product.

Optimal Distance and Optimal Orientation

Distance and *orientation* are popular predicates for describing spatial objects in geographic information system (GIS) settings. In order to use such quantitative predicates, we must specify values for them. For example, we may set 10 kilometers as an upper limit on what we consider to be a short driving distance. We can then formulate queries in terms of this threshold value. However, the implications using such specific values differ from one application domain to the next. Moreover, the specific values themselves may be important spatial knowledge that we need to derive. For these reasons, we considered data mining functions for finding optimal distance and optimal orientation, and developed efficient algorithms for them [13].

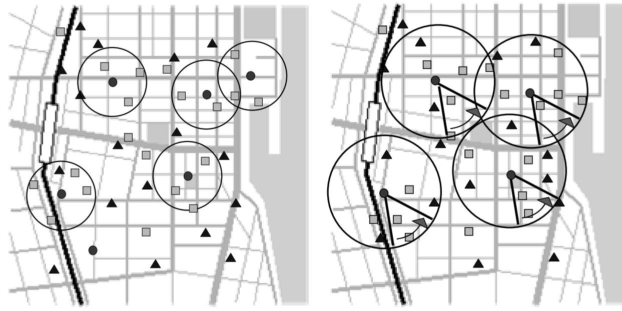


Figure 2. Optimal Distance, Optimal Orientation

The examples of Figure 2 illustrate the use of optimal distance and orientation functions. Here, the concepts associated with each point are indicated by their shapes: **circle** points are related to the concept of "education" (for example, a school or university), **square** points denote the concept of a "profitable branch", and **triangle** points are associated with the concept of an "unprofitable branch". The radius r of the large circles in the left figure is the optimal distance that maximizes the density of profitable branches with distance strictly less than r from some education point, measured as a proportion of all branches (profitable and unprofitable) in the same area. Similarly, the right figure, the sectors indicate the optimal orientation range from the concept, where as before r is the optimal radius within which the proportion of profitable branches is maximized, with the exception that the areas of interest are limited to those points with a fixed orientation with respect to the central education point. By using the insights provided by optimal distance and orientation functions, the hypothetical company of our example can choose an appropriate location at which to open a new branch.

Optimal Region

A geospatial association can also be utilized to compute an optimal connected pixel grid region with respect to some criterion. We first make a pixel grid of an area of interest as shown in Figure 3. For each pixel, we can compute the density of a specific concept. For example, if we are interested in the concept "profitable branch," we could compute the density of point objects associated with "profitable branch." Minimization or maximization of density over a collection of grid regions can lead to significant spatial knowledge discoveries.

Although in general the problem of finding an optimal grid region is NP-hard, if we limit the grid region to

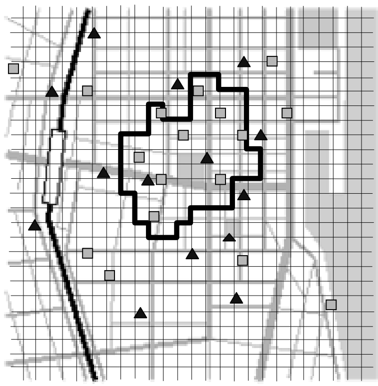


Figure 3. Optimal Region

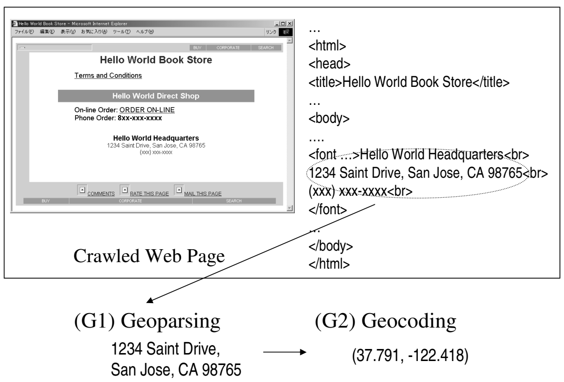


Figure 4. Geospatial Information Extraction

be of x-*monotone* or *rectilinear* shape, we can use efficient algorithms developed for computing optimized two-dimensional rules [6]. With this restriction, we can efficiently compute optimal x-monotone or rectilinear grid regions maximizing the density. In the example of Figure 3, the enclosed region can be considered to be one of high profitability.

2. Mining Geospatial Associations

In this section, we describe how to compute geospatial associations of the form shown in Table 1, from very large collections of web pages obtained by means of a web crawler.

Figure 4 gives a conceptual overview of how our system extracts geospatial information and associates it with web pages. For each web page, we parse the source code of the page and try to find geospatial information `(G1)`. We then translate the geospatial information into coordinate values such as latitude-longitude pairs `(G2)`.

Simultaneously, we extract concepts from the collection of web pages. Figure 5 and 6 provide an overview of the extraction process. From each page, we eliminate `HTML` tags and extract names, terms, abbreviations, and other items

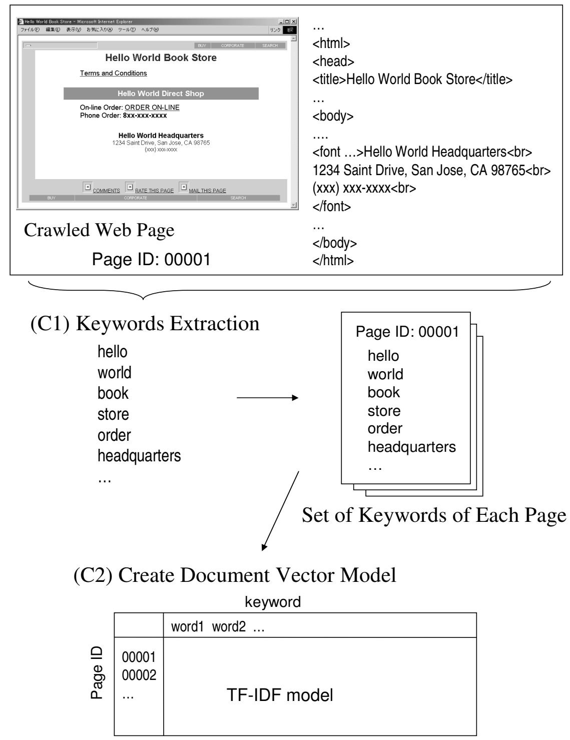


Figure 5. Concepts Extraction (1)

of significance `(C1)`. From the set of extracted items, we create a matrix of document vectors, where each vector corresponds to an individual web page, and each vector attribute corresponds to an item from the total set of extracted items `(C2)`. We use the *term frequency inverse document frequency* (`tf-idf`) model for representing extracted items [10]. Next, we reduce the number of columns of the `tf-tdf` matrix in order to lower computational costs `(C3)`. After this dimensional reduction, clusters of web pages are produced by means of a clustering algorithm applied to the reduced-dimensional matrix `(C4)`. Finally, we label each cluster with several significant keywords indicating concepts associated with its constituent web pages `(C5)`.

2.1. Geospatial Information Extraction

The process of recognizing geographic context is referred to as *geoparsing*, and the process of assigning geographic coordinates is known as *geocoding*. In the field of GIS, various geoparsing and geocoding techniques have been explored and are utilized. In [3] and [11], geoparsing and geocoding techniques are presented that are especially well-suited for web pages. From a large collection of web pages produced by a web crawler, we select those pages containing location information that can be identified using geoparsing techniques, and create lists of the format shown in Table 2, consisting of a web page ID (`Page ID`),

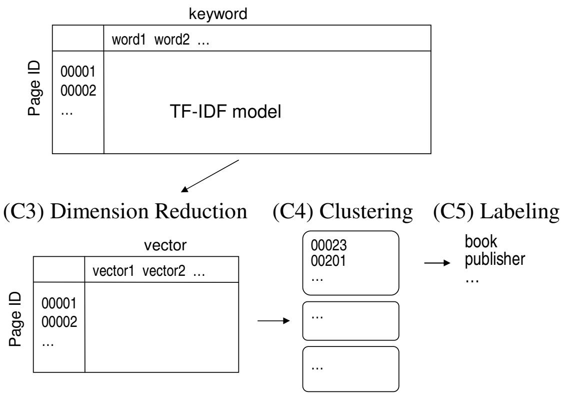


Figure 6. Concepts Extraction (2)

geospatial coordinate values (`Coordinates`), the URL (`URL`), and the title of the page (`Title`).

Table 2. Geocoded Web Pages

ID	Coordinates	URL	Title
1	37.79,-122.41	www.···	Museum of ···
2	37.64,-122.42	www.···	San Jose Book
2	37.78,-122.39	www.···	San Jose Book
	···	···	···

The format of postal addresses varies greatly from one country to another. Moreover, within a given country a variety of expressions may be used. However, within many countries, recognition of postal addresses and zip (postal) codes from natural language text data is well-established. For example, within the United States, we can geocode such location information using the product "Tiger/Zip+4" available with the TIGER dataset [1].

In addition to explicit location information such as addresses, web pages contain other types of information from which we can infer location. Such implicit information can also be utilized to make the lists in Table 2 and to increase the accuracy of location. Phone numbers are an important example of such implicit location information, since they are organized according to geographic principles. IP addresses, hostnames, routing information, geographic feature names, and hyperlinks can also be utilized to infer location [11].

Note that some web pages refer to more than one explicitly recognizable location. Such web pages can therefore be assigned more than one set of coordinate values.

2.2. Vector Space Modeling

We use vector space modeling (VSM) for representing a collection of web pages. Specifically, each web page is represented by a document vector consisting of a set of weighted keywords. Keywords in each web page are extracted using the stemming tool `textract`. `Textract` eliminates HTML tags from each web page, and uses natural language processing techniques to collect significant items from its text content, such as names, terms, and abbreviations [4, 2].

The weights of the entries in each document vector are determined according to the *term frequency inverse document frequency* (`tf-idf`) model. The weight of the i-th keyword in the j-th document, denoted by $a(i,j)$, is a function of the keyword's term frequency $tf_{i,j}$ and the document frequency df_i as expressed by the following formula:

$$a(i,j) = \begin{cases} (1 + tf_{i,j}) \, \log_2 \dfrac{N}{df_i}\,, & \text{if } tf_{i,j} \geq 1, \\ \\ 0\,, & \text{if } tf_{i,j} \geq 1. \end{cases}$$

where $tf_{i,j}$ is defined as the number of occurrences of the i-th keyword w_i in the j-th document d_j, and df_i is the number of documents in which the keyword appears. Once each $a(i,j)$ has been determined, a data set consisting of M web pages spanning N keywords attributes can be represented by an M-by-N matrix $\mathbf{A} = (a_{i,j})$.

The next step is to construct an N-by-N covariance matrix $\mathbf{C}$ from $\mathbf{A}$ as defined below:

$$\mathbf{C} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{d_i} \mathbf{d_i^t} - \overline{\mathbf{d}}\overline{\mathbf{d}}^{\mathbf{t}},$$

where $\mathbf{d_i}$ represents the i-th document vector and $\overline{\mathbf{d}}$ is the average over the set of all document vectors, i.e., $\overline{\mathbf{d}} = [\overline{d}_1 \cdots \overline{d}_N]^{\mathbf{t}}$; $\mathbf{d_i} = [a_{i,1} \cdots a_{i,N}]^{\mathbf{t}}$, and $\overline{d}_j = \frac{1}{M}\sum_{i=1}^{M} a_{i,j}$. Since the covariance matrix is symmetric and positive semi-definite, it can be decomposed into the product $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V^t}$, where $\mathbf{V}$ is an orthogonal matrix which diagonalizes $\mathbf{C}$ so that the sizes of the diagonal entries of $\mathbf{\Lambda}$ are in monotone decreasing order from top to bottom. We can substitute $\mathbf{C}$ with the M-by-n matrix $\mathbf{C}_n$, formed by taking the n eigenvectors corresponding to the largest n eigenvalues of $\mathbf{C}$ [9].

We empirically determined that clusters of web pages can be efficiently computed using values of n of approximately 200. Accordingly, we use this value of n for the target reduced dimension in the experiments reported in Section 3.

2.3. Query-Based Clustering

We compute clusters of web pages from the M-by-n reduced-dimensional matrix. Though the number of web pages M tends to be very large, we used a scalable and effective query-based clustering method suitable for clustering large sets of text data.

In general, a web page may contain several topics in its contents, and thus may contribute to several concepts. Any clustering method based on text data drawn from web pages should take into account the following desiderata:

- An individual data element need not be assigned to exactly one cluster. It could belong to many clusters, or none.
- Clusters should be allowed to overlap to a limited extent. However, no two clusters should have a high degree of overlap unless one contains the other as a subcluster.
- Cluster members should be mutually well-associated. Chains of association whose endpoints consist of dissimilar members should be discouraged.
- Cluster members should be well-differentiated from the non-members closest to them. However, entirely unrelated elements should have little or no influence on the cluster formation.

We will measure the level of mutual association within clusters using *shared neighbor* information, as introduced by Jarvis and Patrick [8]. Shared neighbor information and its variants have been used in the merge criterion of several agglomerative clustering methods [5, 7]. In this section, we show how it can be used as the basis of a definition of cluster integrity.

Neighborhood Patches

Let S be a database of elements drawn from some domain D, modeled using `tf-idf` as described above. We measure the pairwise similarity between two element vectors of D by means of the cosine of the angle between the two vectors, namely

$$\textit{cosangle}(v, w) = \frac{v \cdot w}{\|v\|\,\|w\|}.$$

Here, a value of $\textit{cosangle}(v, w) = 1$ indicates a perfect match between v and w; a value of $\textit{cosangle}(v, w) = 0$ indicates that v and w have no attributes in common.

For every query pattern q drawn from D, let $\mathbf{NN}(S, q, k)$ denote a k-nearest neighbor set of q, drawn from S according to *cosangle*, subject to the following conditions:

- If $q \in S$, then $\mathbf{NN}(S, q, 1) = \{q\}$. That is, if q is a member of the data set, then q is considered to be its own nearest neighbor.
- $\mathbf{NN}(S, q, k-1) \subset \mathbf{NN}(S, q, k)$ for all $1 < k \le |S|$. That is, smaller neighborhoods of q are strictly contained in larger neighborhoods.

The neighbor sets are consistent with a single fixed ranking $\langle q_1, q_2, \ldots, q_{|S|} \rangle$ of the points of S, such that $q_1 = q$, and $\textit{cosangle}(q, q_i) \ge \textit{cosangle}(q, q_j)$ for all $1 \le i < j \le |S|$.

We will sometimes refer to the unique set $\mathbf{NN}(S, q, k)$ as the k*-patch* of q (relative to S), or simply as a *patch* of q.

The proposed methods rely on a parameter that uses shared-neighbor information to assess the mutual association among elements of a patch. We define the *average shared neighbor* score (ASN) of patch $\mathbf{NN}(S, q, k)$ as

$$\mathbf{ASN}(S, q, k) = \frac{1}{k^2} \sum_{v \in \mathbf{NN}(S,q,k)} |\mathbf{NN}(S, q, k) \cap \mathbf{NN}(S, v, k)|.$$

A value of $\mathbf{ASN}(S, q, k) = 1$ indicates perfect mutual association of elements within the patch, and values of $\mathbf{ASN}(S, q, k)$ approaching 0 indicate minimal association.

Shared Neighbor Classification

Consider now the case where the query element q is a member of some cluster within S. Using shared neighbor information, we wish to determine the k-patch based at q that best describes the cluster, over some desired range $a \le k \le b$. The ideal patch would be expected to consist primarily of cluster elements, and to have a relatively high degree of internal association, whereas larger patches would be expected to contain many elements from outside the cluster, and to have a relatively low degree of association. The evaluation focuses on two patches: an *inner* patch of size k indicating a candidate *query* cluster, and an *outer* patch of size proportional to k indicating the local background against which the suitability of the inner patch will be judged.

For a given choice of k, we examine the neighbor sets of each element of the outer patch $\mathbf{NN}(S, q, \lfloor ek \rfloor)$, where the proportion $e > 1$ is chosen independently of k. Consider the neighbor pair (v, w) with v in the outer patch, and w a member of $\mathbf{NN}(S, v, \lfloor ek \rfloor)$. With respect to the inner patch $\mathbf{NN}(S, q, k)$, the pair (v, w) is *conformant* if one of the following two conditions holds.

- *Internal pair*: v and w both lie in the inner k-patch, and w is shared by the k-patches of both q and v.
- *External pair*: v and w do not both lie in the inner k-patch, and w is not shared by the $\lfloor ek \rfloor$-patches of both q and v.

In the internal case, an association between q and inner patch member v is recognized by virtue of their shared neighbor w also belonging to the inner patch. In the external case, if an association between q and v cannot be indicated within the inner patch by w, then w does not indicate an association between q and v within the outer patch — that is, the larger patch offers no advantages over the smaller with respect to the neighbor pair (v, w).

Ideally, the k-patch best describing the cluster containing q would achieve a high proportion of conformant neighbor pairs, both internal and external. A high proportion of conformant internal pairs indicates a high level of association within the k-patch, whereas a high proportion of conformant external pairs indicates a high level of differentiation with respect to the local background. As both considerations are equally important, these proportions should be accounted for separately. We achieve this by maximizing, over all choices of k in the range $a \leq k \leq b$, the sum of the proportions of internal and external conformant neighbor pairs.

Boundary Sharpness Maximization

The proportion of internal conformant neighbor pairs is given by

$$\frac{\sum_{v \in \mathbf{NN}(S,q,k)} |\mathbf{NN}(S,v,k) \cap \mathbf{NN}(S,q,k)|}{k^2} = \mathbf{ASN}(S,q,k);$$

the proportion of external conformant neighbor pairs can be shown to be

$$1 + \frac{k^2 \mathbf{ASN}(S,q,k)}{\lfloor ek \rfloor^2 - k^2} - \frac{\lfloor ek \rfloor^2 \mathbf{ASN}(S,q,\lfloor ek \rfloor)}{\lfloor ek \rfloor^2 - k^2}.$$

The *boundary sharpness maximization* (BSM) problem can thus be formulated as follows:

$$\max_{a \leq k \leq b} \mathbf{BSV}(S,q,k,e),$$

where

$$\mathbf{BSV}(S,q,k,e) = \frac{\lfloor ek \rfloor^2 \left[\mathbf{ASN}(S,q,k) - \mathbf{ASN}(S,q,\lfloor ek \rfloor)\right]}{\lfloor ek \rfloor^2 - k^2}$$

shall be referred to as the *boundary sharpness* of the k-patch $\mathbf{NN}(S,q,k)$ with respect to e.

Patch Profiles

Although at first glance it would seem that boundary sharpness values are expensive to compute, with a careful implementation the costs can be limited to $O(e^2b^2)$ time. This is achieved through the efficient computation of a *profile* of values of $\mathbf{ASN}(S,q,k)$ for $1 \leq k \leq eb$. Patch profiles are useful not only for the automatic determination of boundary sharpness values, but also (when plotted) as an effective visual indicator of the varying degrees of association within the neighborhood of a query element. Examples of patch profiles appear in Figures 7 and 8.

Total Clustering

Given a target range of cluster sizes $a \leq k \leq b$, the following simple strategy greedily selects elements of S to be the bases of query clusters. It makes use of a user-supplied threshold α on the maximum allowable proportional overlap between two clusters.

1. For each $v \in S$, compute $k(v)$ maximizing $\mathbf{BSV}(S,v,k,e)$ over the range $a \leq k \leq b$.

2. Initialize the set of candidate query cluster bases to $C = S$.

3. Select the candidate $q \in C$ with largest boundary sharpness value $\mathbf{BSV}(S,q,k(q),e)$ to be the basis of a query cluster, namely $\mathbf{NN}(S,q,k(q))$.

4. Eliminate from C all other candidates v whose query cluster has substantial overlap with that of q. In particular, v is eliminated if

$$\frac{|\mathbf{NN}(S,v,k(v)) \cap \mathbf{NN}(S,q,k(q))|}{|\mathbf{NN}(S,v,k(v))|} \geq \alpha.$$

5. Repeat Steps 3–4 until C is exhausted.

Many variants of this strategy can be envisioned; in particular, larger ranges of sample cluster sizes can be efficiently accommodated by applying the method to random samples drawn from S. Also, a minimum threshold δ can be set on the boundary sharpness values of reported clusters.

2.4. Cluster Labeling

We can assign labels to a cluster based on a ranked list of terms that occur most frequently within the web pages of the cluster, in accordance with the term weighting strategy used in the document vector model. Each term can be given a score equal to the sum or the average of the corresponding term weights over all document vectors of the clusters; a predetermined number of terms achieving the highest scores can be ranked and assigned to the cluster.

When dimensional reduction is used, the original document vectors may no longer be available due to storage limitations. Nevertheless, meaningful term lists can still be extracted without the original vectors. The i-th term can be associated with a unit vector $z_i = (z_{i,1}, z_{i,1}, ..., z_{i,N})$ in the original document space, such that $z_{i,j} = 1$ if $i = j$, and $z_{i,j} = 0$ otherwise. Let ϕ be the average of the document vectors belonging to the query cluster $\mathbf{NN}(R,q,k)$. Using this notation, the score for the i-th term can be expressed as $z_i \cdot \phi$. However, since $\|z_i\| = 1$ and ϕ is a constant, ranking the terms according to these scores is equivalent to ranking them according to the measure

$$\frac{z_i \cdot \phi}{\|\phi\|} = \textit{cosangle}(z_i, \phi).$$

With dimensional reduction, the pairwise distance *cosangle*(v,w) between vectors v and w of the original space is approximated by *cosangle*(v',w'), where v' and w' are the respective equivalents of v and w in the reduced dimensional space. Hence we could approximate *cosangle*(z_i, ϕ) by *cosangle*(z'_i, ϕ'), where z'_i and ϕ' are the reduced-dimensional counterparts of vectors z_i and ϕ, respectively. The value *cosangle*(z'_i, ϕ') can in turn be approximated by *cosangle*(z'_i, ϕ''), where ϕ'' is the average of the reduced-dimensional vectors of the query cluster. Provided that the vectors z'_i have been precomputed for all $1 \leq i \leq n$, a ranked set of terms can be efficiently generated by means of a nearest-neighbor search based on ϕ'' over the collection of reduced-dimensional term vectors $\{z'_i | 1 \leq i \leq n\}$.

3. Experimental Results

We performed our experimentation on a set of web pages containing location information identifying them as pertaining to California's Silicon Valley area. This data set (which we will refer to as *CA North*) consisted of $M = 4493$ web pages from which $N = 12,880$ keywords were extracted. We applied the dimension reduction method to reduce the dimension to $n = 200$. We applied the query-based clustering method and labeling method, with the following additional implementation choices:

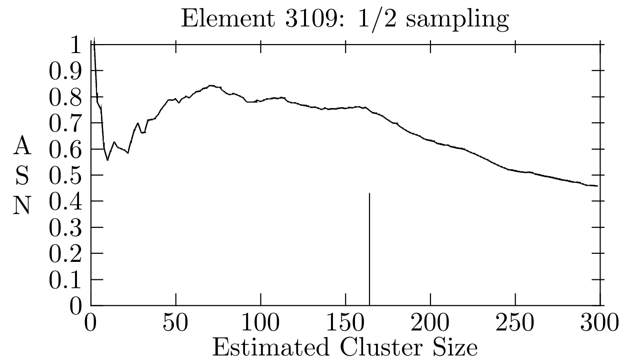
- An overlap proportion of $\alpha = 0.5$ for elimination of duplicate clusters.
- A minimum boundary sharpness threshold of $\delta = 0.15$.
- A cluster size range of $25 \leq k \leq 120$.
- An outer patch size of $\lfloor ek \rfloor = \min\{2k, 150\}$, corresponding to variable choices of e in the range $1.2 \leq e \leq 2$.

The clustering method was applied to uniform random samples drawn from S, of size $\frac{|S|}{2^i}$ for $0 \leq i \leq 3$.

Figure 7 shows an example of clusters produced by the method. The graph shows the relationship between the ASN scores and patch sizes of the cluster. The cluster contains an estimated 164 web pages, and its top terms are "java," "countries.copyright," and others related to the Java programming language. The number associated with each term is the *cosangle* ranking score of the term.

Figure 8 shows detailed information concerning another cluster devoted to "genome." The list shows the titles of 70 nearest web pages from the query element at which the cluster is based. As the estimated size of the cluster is 42, the first 42 titles can ideally be expected to relate to the concept of "genome", whereas the remaining titles should relate to other concepts. Note that the proportion of web pages related to "genome" is much higher inside the cluster than outside, even though there are some misclassified pages in both cases.

From the CA North clusters, we created a geospatial association and applied a neighboring class set function. We found that web pages about "non-commercial" and web pages about "font" refer to locations in close proximity to one another. Similarly, "software" and "telephone" form a frequent neighboring class set.



0.5529199	java
0.5398724	countries.copyright
0.4789391	v1.3
0.4783071	workarounds
0.4528790	method
0.4508706	class
0.4367413	jdbc
0.4202253	2d
0.3722224	constr
0.3643397	summary

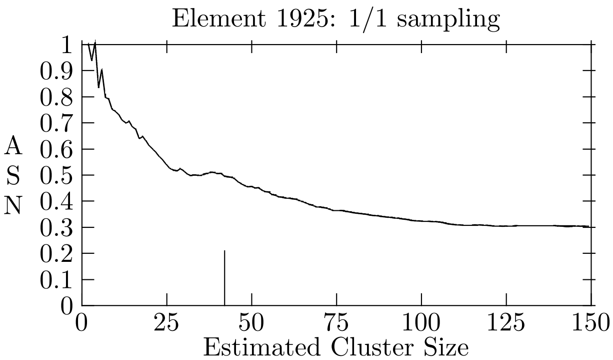
Figure 7. Average shared-neighbor scores and BSM query clusters for an element drawn from the CA North data set.

4. Conclusion

We presented a system for extracting spatial knowledge from collections of web pages containing location information. For each item of location information, we apply geocoding techniques to compute geographic coordinates. Next, we extract significant keywords from web pages to serve as concept descriptors. We associate the keywords with web pages that contain location information to create a geospatial association table. We can find spatial patterns from the geospatial association table by applying spatial data mining techniques.

In the geospatial association tables generated from web pages, false positive records are possible; that is, some pages may contain location information items that are unrelated to the keyword concepts expressed. Such false positive records lead to incorrect spatial knowledge. We found many examples that seem to be false positive records in the experimental results. Further investigation is therefore needed to refine the spatial insights found from web pages.

The typical causes of false positive records are: (1), the many portal web pages containing large lists of addresses with miscellaneous contexts; and (2), the many pages quoting addresses that are not directly related to the main topic of the pages. Web pages are too numerous, too large, and



0.7459853	genome
0.7135191	cdna
0.7100353	gene
0.7068095	sequence
0.7014005	pcr
0.6904713	dna
0.6505522	chromosome
0.6437061	continuum
0.6426465	human genome project
0.6379184	fragment

Rank	*Title*
1 – 2	⟨blank title⟩
3	Lin-7 may be required for Let-23 receptor localization during vulval development.
4 – 5	Distinctive Gene Expression Patterns in Human Mammary Epithelial Cells and . . .
6	Roundtable Discussion: Gene Therapy Progressing Even As Regulators . . .
7	Taipan Webhead June 1998
8	Erik Sandelin, Curriculum Vitae
9	Article
10	BioMEMS 2001 Nanofabrication and Analytical Techniques for Biomedical . . .
11	BRAIN INJURY ADVOCATES – Law Offices of Harvey A. Hyman
12	AFGC Website
13	High Capacity Shaker – Technology Development Center
14	⟨blank title⟩
15	CELL Tools, Inc.
16	Research Abstracts 2000 DOE Human Genome Program
17	Memes, Meta-memes and Politics
18	Erstwhile darling of Wall Street, Incyte Pharmaceuticals Inc
19	IGeneX Inc, – References
20 – 21	Plant Genome I Abstracts
22	Untitled Document
23	⟨blank title⟩
24	The Wavelet Digest – Volume 4, Issue 4 (April 12, 1995)
25	Miles Lab: Available Positions
26	Biocompare – The Buyer's Guide for Life Scientists – Browse Products
27	Removing the Time Axis . . . : Neural Networks versus Memory-Based Machine Learning
28	Directed Evolution and Solid Phase Enzyme Screening
29	Genome Mapping Research Abstract Index
30	SFSU Biology Department – Burrus Lab
31	Diversa Corporation: Press Releases for 2000
32	Poster Images
33	CV Therapeutics/Company Information
34	AJP: Journal of Neurophysiology Table of Contents
35	New York Times: Personal Health: Preventing Glaucoma
36	Supplemental
37 – 39	Research Abstracts 2000 DOE Human Genome Program
40	AJP: Lung Cellular and Molecular Physiology Table of Contents
41	Project Inform's Anabolic Steroids [HIV / AIDS Treatment Information]
42	Sonomorph Music 93
43	CI CW 10.32
44	. . . Newsletter of the Molecular Manufacturing Shortcut Group of the National Space Society
45	AJP: Renal, Fluid and Electrolyte Physiology Table of Contents
46	The David and Lucile Packard Foundation
47	Mixed Signal Integration – Contact
48	Chronic Illness and Independence
49	sand0999 No Ads
50	LabOne Script (Referral Form)
51	AJP: Journal of Neurophysiology Table of Contents
52	"DOCK00 User Group Meeting"
53	DoubleTwist, Inc.
54	A VirtualPBX.Com Usage Scenario: International Offices World Wide Connected by . . .
55	Poster 115
56	AJP: Journal of Neurophysiology Table of Contents
57	News/Spectrum/Cellular
58	⟨blank title⟩
59	The Risks Digest Volume 13: Issue 1
60 – 61	Labs on the Net
62	Generic Biopharmaceuticals: Early Players Shape the Rules
63	Bibliography of Cetacean Releases
64	Microcurrent: Electrotherapy with Microcurrent
65	INCOSE Speakers Bureau
66	ACNielsen Vantis – Career Opportunities
67	Jennifer Widom's home page
68	Monster Cable – Robert Harley
69	Has Penrose Disproved A.I.?
70	Download: White Papers, Embedded Intelligence

Figure 8. Details of a BSM query cluster of estimated size 42.

too unstructured to allow the custom annotation of each page according to the use and relevance of the spatial information quoted. We must therefore consider as an important direction for future research the development of more efficient and accurate geoparsing methods for generating geospatial associations.

Acknowledgment

The authors thank Dr. James W. Cooper and his project members at IBM T. J. Watson Research Center for providing the text mining functions of `textract` for extracting significant terms from web pages.

References

[1] http://www.census.gov/geo/www/tiger/.

[2] B. Boguraev and M. S. Neff. Discourse segmentation in aid of document summarization. In *Proc. of the Hawaii International Conference on System Sciences (HICSS)*, 2000.

[3] O. Buyukokkten, J. Cho, H. Garcia-Molina, L. Gravano, and N. Shivakumar. Exploiting geographical location information of web pages. In *Proc. of Workshop on Web Databases (WebDB)*, 1999.

[4] R. Byrd and Y. Ravin. Identifying and extracting relations in text. In *Proc. of the Applications of Natural Language to Information Systems (NLDB)*, pages 149–154, 1999.

[5] L. Ertoz, M. Steinbach, and V. Kumar. Finding topics in collections of documents: A shared nearest neighbor approach. Technical Report Preprint 2001-040 (8 pages), University of Minnesota Army HPC Research Center, 2001.

[6] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining with optimized two-dimensional association rules. *ACM Trans. on Database Systems*, 26(2):179–213, 2001.

[7] S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000.

[8] R. A. Jarvis and E. A. Patrick. Clustering using a similarity measure based on shared nearest neighbors. *IEEE Transactions on Computers*, (11):1025–1034, 1973.

[9] L. Malassis and M. Kobayashi. Statistical methods for search engines. Technical Report RT-413 (33 pages), IBM Tokyo Research Laboratory Research Report, 2001.

[10] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2000.

[11] K. McCurley. Geospatial mapping and navigation of the web. In *Proc. of World Wide Web (WWW)*, pages 221–229, 2001.

[12] Y. Morimoto. Mining frequent neighboring class sets in spatial databases. In *Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data mining (KDD)*, pages 353–358, 2001.

[13] Y. Morimoto, H. Kubo, and T. Kanda. Mining optimized distance and / or orientation rules in spatial databases. Technical Report RT0404, IBM Tokyo Research Laboratory Research Report, 2001.