Learning Ranking Functions for Geographic Information Retrieval Using Genetic Programming

You-Heng Hu and Linlin Ge

School of Surveying and Spatial Information Systems
University of New South Wales
Sydney, Australia
Phone: +61-2-9385 4185
Fax: +61-2-93137493
Emails: you-heng.hu@student.unsw.edu.au, l.ge@unsw.edu.au

Geographic Information Retrieval (GIR) has emerged as a new and promising tool for representation, storage, organisation of and access to geographic information. One of the current issues in GIR research is ranking of retrieved documents by both textual and geographic similarity measures. This paper describes an approach that learns GIR ranking functions using Genetic Programming (GP) methods based on textual statistics and geographic properties derived from documents and user queries. Our proposed approach has been applied to a large collection of geographic metadata documents. The experimental results show that the ranking functions learned using our method achieved significant improvement over existing ranking mechanisms in retrieval performance.

Keywords: Geographic Information Retrieval; Ranking; Evaluation; Genetic Programming; Geographic metadata

ACM Classifications: H.3 Information Storage and Retrieval; H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval

1. INTRODUCTION

Geographic Information Retrieval (GIR) is a new branch of Information Retrieval (IR) that takes into account geographic context of information when searching and retrieving of documents. Compared with conventional IR systems, GIR systems place the main emphasis on geographic information, which can be defined as information that reference some part of the Earth's surface (Cai, 2002; Larson, 1996). The basic goal of GIR is to help users to find relevant documents from a larger collection based on user queries that in general consist of both thematic criteria and geographic criteria. Examples of GIR queries are "Find news stories about bushfire in Sydney, Australia" and "Find maps of the nearest Chinese restaurant". Given the fact that all human activities are associated with geographic context, GIR has become an important tool for many application areas, such as digital libraries (Woodruff and Plaunt, 1994), online bibliographic systems (Coleman *et al*, 2002) and Internet search engines (Jones *et al*, 2002).

Ranking is one of the key research questions in Information Retrieval. Given a user query, IR ranking functions assign each retrieved document a numerical score, which reflects the relevance of

Manuscript received: 2 June 2007
Communicating Editor: Phoebe Chen

the document to the query. Ranking scores are then used by IR systems to order the retrieved documents, and documents that are most relevant to the query are presented to users first. With the ranked results users can find information of interest more easily. In addition, most of the current evaluation methods for IR system retrieval performance are based on ranked results.

In GIR, the relevance of a document to a user query is determined by their thematic context, and also geographic context. This important characteristic of GIR suggests two distinct but interrelated hypotheses: firstly, geographic features of documents and user queries should be used for the calculation of geographic ranking scores; and secondly, geographic ranking scores and thematic ranking scores should be combined to produce the final ranked result.

Thematic ranking methods assign scores to documents based on lexical and syntactic statistics of documents and queries. On the other hand, geographic ranking methods assign scores to documents based on geographic features discovered from documents and user queries. The main issue in the integration of several ranking approaches into one ranking function is how different measures can be combined. Many integration schemes have been proposed in the literature, which can be divided into three categories: standard combinations, linear combinations, and non-linear combinations. Standard combinations calculate the maximal, minimal or median values of individual ranking results as final results. Linear combinations assign a weight value to each individual ranking result, and take the sum of weighted ranking scores as final results. The performance of standard combinations and linear combinations are studied extensively using large unstructured text collections by Fox and Shaw (1994), Lee (1997) and Vogt and Cottrell (1998). The construction of weight matrices used in linear combinations can be automated using machine learning (Bartell, Cottrell and Belew, 1994) and logistic regression (Gey, 1994). In contrast to standard and linear combinations, non-linear combinations allow individual ranking results to be combined in more complex manners. Neural network and evolutionary computation are two important approaches for the learning of non-linear ranking functions (Bartell, 1994; Trotman, 2005). Many experimental results show that significant improvements on retrieval performance can be achieved by combining multiple ranking strategies. However, there is little previous work on the integration of geographic ranking mechanisms with textual ranking mechanisms for GIR systems, which is the main focus of this paper.

This paper reports our experiences in attempting learning optimal GIR ranking functions for a given document collection using Genetic Programming (GP). GP is an evolutionary computation technique that aims to automatically find an optimised computer program for a specified problem using the principle of natural evolution process (Koza, 1992). The retrieval performance of our proposed approach has been compared with traditional ranking methods, including: classic ranking method based on the Vector Space Model (VSM), simple linear combination of multiple ranking scores, and linear combination of multiple ranking scores using the logistic regression. The results show that our method can achieve significant improvement over these methods. In addition, this paper highlighted issues inherent in the design and implementation of GP-based GIR ranking functions. The impact of different fitness functions and evolution strategies on retrieval performance has been examined. The test collection used in our experiments consists of 4,000 geographic metadata records collected from the New South Wales Natural Resources Data Directory, a state government node of Australian Spatial Data Directory (ASDD). The evaluation method used for retrieval performance comparison was based on the precision and recall measurements of ranked results.

The contributions of this paper are as follows. Firstly, it describes the overall design of a GIR ranking function learning method using GP; secondly, it highlights important decisions regarding GP evolution strategies; thirdly, it proposes and compares four different fitness functions for the GP algorithm, and finally it reports experimental results to support our approach.

The remaining sections of this paper are structured as follows. Section 2 gives some necessary background knowledge. Section 3 describes our methodology for learning GIR ranking functions using GP. Section 4 reports the experiments and results. Section 5 concludes and discusses some directions for future research.

2. BACKGROUND

This section reviews some background knowledge related to this work, including theoretical models of GIR and the Genetic Programming method.

2.1. Theoretical Models of GIR

According to the definition given by Baeza-Yates and Ribeiro-Neto (1999), an information retrieval model specifies aspects like the representations of documents, the representations of user queries, the modelling of relationships between documents and user queries, and the rank mechanisms. This subsection describes theoretical models that are related to GIR, including the geographic model, the Vector Space Model and the GeoVSM model.

2.1.1 Geographic Model

In Geographic Information Science (GISci), geographic features of information are represented as geometric objects (e.g. points, lines and polygons) in a Euclidean space, and other features are represented as attributes of these geometric objects (Tomlin, 1990). User queries in the geographic model are expressed using structured query languages, in which query criteria (geographic and non-geographic) are well specified and are linked using predefined predicates. Geographic criteria of queries are processed based on geometric computation, and non-geographic criteria are processed by matching with attribute values. Ranking in geographic models is based on geometric calculations such as distance and area of overlap between user queries and retrieved documents. The basic hypothesis underlying this method is that the relevance between a document and a user query increases or decreases with an increase or decrease in the geographic distance and/or overlap area between them (Beard and Sharma, 1997; Hill, 1990; Larson and Frontiera, 2004).

2.1.2 The Vector Space Model

The Vector Space Model (VSM) (Salton, 1971) is one of the most popular models in IR. In VSM, both documents and user queries are represented as n-dimensional vectors, where n is the total number of index terms in the system. Each component of a VSM vector is a weighting value associated with an index term and a document (or a query). The similarity between a document and a query is calculated by the cosine of the angle between the document vector and the query vector. The Term Frequency - Inverse Document Frequency (tf-idf) scheme (Salton and Buckley, 1988) is widely used in VSM-based systems for weighting value calculation.

2.1.3 The GeoVSM Model

The GeoVSM model (Cai, 2002) is an attempt to integrate the coordinate-based geographic model with the textual keyword-based Vector Space Model for GIR. In GeoVSM, Each document (or a user query) is represented as a joint of two complementary and non-redundant subspaces, one is a geographic subspace using the geographic model and another one is thematic subspace using the Vector Space Model. The two subspaces are internally connected using document identifications. To measure the degree of relevance of a document to a query, GeoVSM calculates geographic similarity (SimG) and thematic similarity (SimT) separately and then combines the two relevance measures to a single ranking score, as shown in the following:

$$sim(d,q) \quad = f(SimT(d,q), SimG(d,q))$$

2.2. Genetic Programming

Genetic programming (GP) is a sub-field of evolutionary computation technique that aims to automatically find an optimised computer program for a specified problem using the principle of natural evolution process. GP theory has its root in the classical Genetic Algorithm (GA) theory, but GP provides a more expressive way to represent the search space and solutions. In GP literatures, the search space is called population and a solution is called an individual. GP algorithms utilise genetic operators such as reproduction, crossover and mutation on each generation of individuals to produce new generations of better solutions.

The implementation of a GP algorithm consists of three elements:

1. A set of terminals and functions that can be as logic unit of a computer program. The terminal collection consists of independent variables of the problem. The function collection consists of all operations that meaning for the problem, such as arithmetic operators (i.e. add, subtract, multiply and divide), mathematic functions (e.g. log, square root and circular functions) and logical operations (e.g. AND, OR and NOR). Each individual is represented as a tree, in which inner are selected from the function collection and the leaf nodes are selected from the terminal collection. The mathematic formula of an individual can be represented using a tree structure.
2. A fitness measure evaluates how well each individual in the population is for the problem. Both absolute and relative measures can be used for fitness evaluation. For absolute measures, the fitness value of an individual is calculated based on a comparison with an optimal solution. Absolute measures are used widely in applications like regression model evolving (Eggermont and van Hemert, 2001). Using relative measures, the fitness value is calculated based on a comparison with results of other individual in the population. Relative measures are used for many more complicated problems like the artificial ant problem described by Jefferson (1990), in which the optimal solution cannot be given using well-defined mathematical models. The selection and design of fitness evaluation functions are one of the key issues in a successful implementation of GP algorithms because the fitness value of an individual decides the probability that it would be selected for genetic operations.
3. An evolution strategy specifies control mechanisms of GP evolution process. The most important decision choices are: (a) how the first generation is created, related control parameters include the population size, and the methods for selection tree nodes from function and terminal sets; (b) the genetic operators used in the evolution and the probabilities of each operator, which decide the frequencies of using these operators; (c) how individuals are selected as parents for the next generation. Selection methods can use raw fitness values of individuals, and also can use ranking order of individuals based on their fitness values; and (d) conditions under which the evolution will be terminated, for example, the maximum number of generations is achieved. The selection of evolution strategies has direct impact on complexity of the GP implementation.

3. METHODOLOGY

This section describes our methodology for learning GIR ranking functions for a given document collection and a set of queries using GP. There are several reasons that GP is chosen for this task: firstly, GP has been demonstrated to be capable of evolving complex program structures and giving solutions to some problems that are competitive with human-written algorithms. Secondly, GP can be used for solving both linear and non-linear problems, and finally, previous work in conventional IR systems has shown that ranking functions learned using GP achieve significant improvement in retrieval performance (Fan, Gordon and Pathak, 2005).

Name	Description
DOC_COUNT	Number of documents in the collection
RAWFREQ_KEYWORD	Raw frequency of the keyword in the document
DOCFREQ_KEYWORD	Number of documents in which the keyword appears
RAWFREQ_PLACE	Raw frequency of the place name in the document
DOCFREQ_PLACE	Number of documents in which the place name appears
MAX_RAWFREQ	The maximum raw frequency of all keywords and place names in the document
OVERLAP_AREA	Area size of overlap between the document and the query
DOC_AREA	Area of the document
QUERY_AREA	Area of the query
DOC_OVERLAP	Number of documents which overlap the query

Table 1: Terminals used in the GP learning process

3.1. Terminals and Functions

Terminals and functions are logic units of GP individuals. Terminals reflect logical views of documents and user queries. In our method, index terms (i.e. keywords) are used to describe the textual content of documents, and place names and geometric objects (i.e. a polygon) are used to describe geographic features of a document. User queries are represented using the same way. Based on such representation, those listed in Table 1 are selected as GP terminals.

Terminals listed in Table 1 can be categorised into two groups: local and global. The local data reflects content of one particular document. *RAWFREQ_KEYWORD*, *RAWFREQ_PLACE*, *MAX_RAWFREQ*, *OVERLAP_AREA*, *DOC_AREA* and *QUERY_AREA* are examples of local data. In contrast, global data reflects content of the whole collection. *DOC_COUNT*, *DOCFREQ_KEYWORD*, *DOCFREQ_PLACE* and *DOC_OVERLAP* are examples of global data.

Functions reflect the relationships between terminals. Functions used in our experiments include addition (+), subtraction (-), multiplication (×), division (/) and natural logarithm (log). Additional controls are added to the function definitions to handle exception cases, such as divided by zero, and logarithm of non-positive numbers.

Each GP individual is represented as a tree, in which inner nodes are selected from the function collection and leaf nodes are selected from the terminal collection. Figure 1 shows an example of GP individual whose mathematic formula is $\frac{x + y}{x^2 + y^2}$.

3.2. Fitness Functions

Fitness functions reflect how well each individual is. A good design of fitness functions can help to reduce learning time and to produce better solution. Four fitness functions are considered in our method:

Fitness1. The first fitness function utilises the harmonic mean of standard precision and recall measures.

$$Fitness1 = \frac{1}{Q} \times \sum_{i=1}^{Q} \frac{2}{1/P_i + 1/R_i} \tag{1}$$

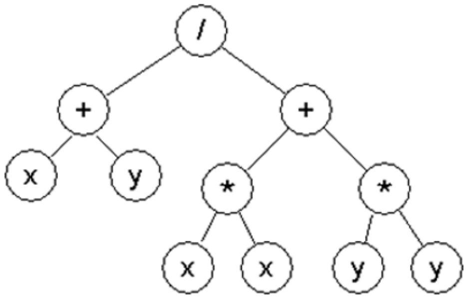


Figure 1: An example of the tree representation of GP individuals

where Pi and Ri are the precision and the recall value of the ith query respectively, Q is the total number of queries. This function assigns a high fitness value to a solution when both recall and precision are high. The advantage of this function is its mathematical simplicity. However, the main disadvantage of this function is that the ranking order of retrieved results is not taken into account.

Fitness2. The second fitness function returns the arithmetic mean of the precision values at 50% recall for all queries as results.

$$Fitness2 = \frac{1}{Q} \times \sum_{i=1}^{Q} P_{i,50} \tag{2}$$

where Q is the total number of queries, Pi,50 is the precision value at 50% recall level of the ith query. The advantage of this function is that it takes into account the order of retrieved result, and higher fitness values are assigned to solutions that retrieve relevant documents quickly.

Fitness3. The third fitness function utilises the idea of average precision at seen relevant documents.

$$Fitness3 = \frac{1}{Q} \times \sum_{i=1}^{Q} (\frac{1}{D_i} \times \sum_{j=1}^{D_j} \left(r(d_j) \times \frac{\sum_{k=1}^{j} r(d_k)}{j} \right)) \tag{3}$$

where Q is the total number of queries, Di is the total number of retrieved documents for the ith query, and r(dk) is a function that returns 1 if the document dk is relevant to the ith query and returns 0 otherwise.

Fitness4. The last fitness function is of our own design, which utilises the weighted sum of precision values on the 11 standard recall levels.

$$Fitness4 = \frac{1}{Q} \times \sum_{i=1}^{Q} (\sum_{j=0}^{10} \frac{1}{(j+1)^m} P_{i,j}) \tag{4}$$

where Q is the total number of queries, Pi,j is the precision value at the jth recall level of the 11 standard recall levels for the ith query, m is a positive scaling factor determined from experiments. This fitness function prefers systems that have high precision values at low recall level. Fitness4 is believed to be a good one as precision values at all standard recall value contribute to the final results, and the scaling factor m can be tuned for different document collections and query sets.

3.3. Genetic Operators

The evolution procedure in GP can be described as a repeated procedure that creates a new generation by applying various genetic operators on the previous generation. Three genetic operators are used in our method to create new generations, including: creation, crossover and reproduction.

The creation operator creates a new individual as a random selection procedure, which assumes each terminal and function has the same probability of being selected. A node is first selected and is added to the tree that presents the solution as the root node. If the selected node is a terminal, the procedure is finished. If the selected node is a function, one or more sub-trees need to be created. The number of sub-trees the function node has is decided by the function definition. For example, the addition, subtraction, multiplication and division functions have two sub-trees, because they need two input parameters; the natural logarithm function has one sub-tree, because only one input parameter is required. Sub-trees are recursively created using the same method until no sub-tree is required. Using the creation operator, the produced individual added to the new generation increases the diversity of the population.

The crossover operator first selects two individuals as parents, and randomly selects a sub-tree from each parent. Then two new individuals are generated by swapping the two sub-trees of each parent. The crossover operator is one of the most widely used genetic operators in GP, because it creates new individuals faster than other operators, and it creases the diversity of individuals in the new generation. Figure 2 illustrates how the crossover operator works.

The reproduction operator selects one individual and copies it into the new generation directly without any modification. This operator doesn't improve the diversity of new generation, but it is helpful to keep good individuals during the evolution.

4. EXPERIMENTS

A set of experiments was conducted to evaluate the effectiveness of learning ranking functions using GP for GIR. Constants used in our experiments include the maximum number of generations

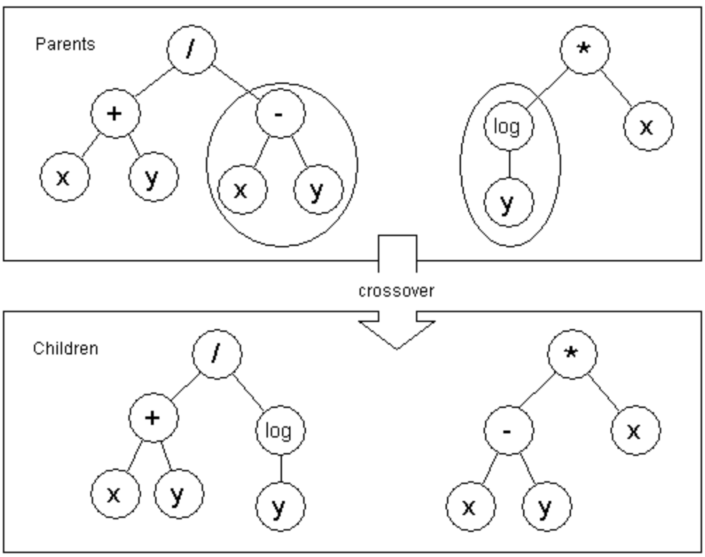


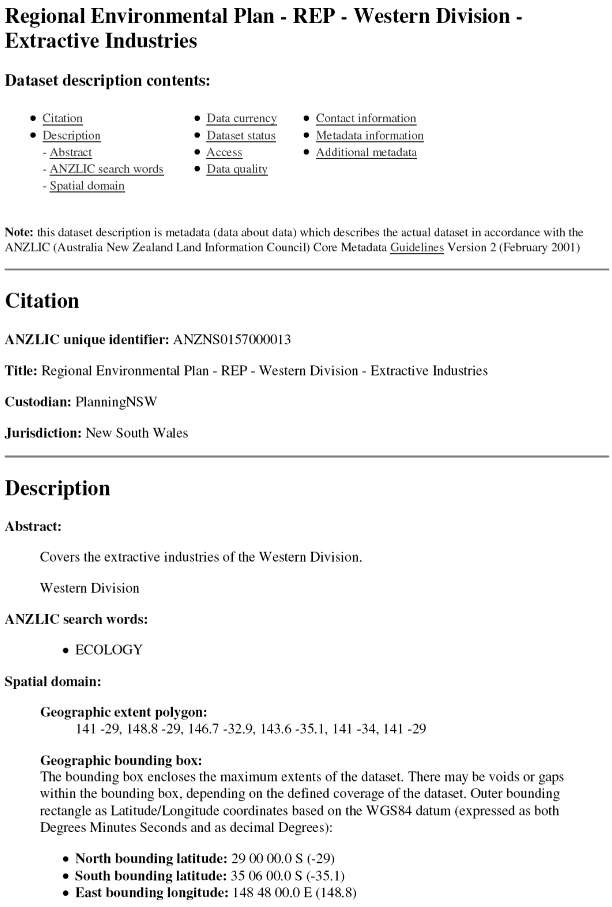
Figure 2: The crossover operation

that was set to 50, and the maximum population size that was set to 100. These values were determined from experiments based on suggested value ranges given by De Jong (1975).

Because of the randomness of the GP evolution, each experiment was run three times and the best result of the three runs is selected as the final result. The experiments were implemented using JAVA programming language, and were run on a 2.8 GHz processor Linux machine with 512M memory.

4.1. Data

The document collection used in our experiments consists of 4,000 geographic metadata records collected from the New South Wales Natural Resources Data Directory, a state government node of Australian Spatial Data Directory (ASDD). The original document collection was split into two subsets, training data (75%, 3000 documents) and test data (25%, 1000 documents). 100 queries were generated by randomly selecting from a large pool for both training and testing. Each query consists of three fields: keyword, place name and the minimum-bounding rectangle (MBR) of the place. Binary relevance judgments (i.e. a document is relevant to a query or not) were made by human judges for each query. Figure 3 and Figure 4 shows an example of documents and queries used in the experiments.



Regional Environmental Plan - REP - Western Division - Extractive Industries

Dataset description contents:

- Citation
- Description
  - Abstract
  - ANZLIC search words
  - Spatial domain
- Data currency
- Dataset status
- Access
- Data quality
- Contact information
- Metadata information
- Additional metadata

Note: this dataset description is metadata (data about data) which describes the actual dataset in accordance with the ANZLIC (Australia New Zealand Land Information Council) Core Metadata Guidelines Version 2 (February 2001)

Citation

ANZLIC unique identifier: ANZNS0157000013

Title: Regional Environmental Plan - REP - Western Division - Extractive Industries

Custodian: PlanningNSW

Jurisdiction: New South Wales

Description

Abstract:

Covers the extractive industries of the Western Division.

Western Division

ANZLIC search words:

- ECOLOGY

Spatial domain:

Geographic extent polygon:
141 -29, 148.8 -29, 146.7 -32.9, 143.6 -35.1, 141 -34, 141 -29

Geographic bounding box:
The bounding box encloses the maximum extents of the dataset. There may be voids or gaps within the bounding box, depending on the defined coverage of the dataset. Outer bounding rectangle as Latitude/Longitude coordinates based on the WGS84 datum (expressed as both Degrees Minutes Seconds and as decimal Degrees):

- **North bounding latitude:** 29 00 00.0 S (-29)
- **South bounding latitude:** 35 06 00.0 S (-35.1)
- **East bounding longitude:** 148 48 00.0 E (148.8)

Figure 3: An example document used in the experiments

```
<QUERY>
<keyword>BOUNDARIES</keyword>
<place>NEW SOUTH WALES</place>
<bounding>-28.150112;-37.512672;140.997864;159.101898</bounding>
</QUERY>
```

Figure 4: An example query used in the experiments

4.2. Baselines for Retrieval Performance Evaluation

Three exiting ranking methods were used as baselines for comparison of the experimental results and for evidence of the effectiveness of the ranking function learned by GP.

Jakarta Lucene Search Engine (Lucene): Jakarta Lucene (c.f. http://lucene.apache.org) is a high performance full text search engine, in which the ranking function is based on the VSM and the tf-idf scheme.

Linear combination (LC): This ranking function assigns a ranking score to each retrieved document by linearly combining the thematic similarity and the geographic similarity measures between a document and a query. The ranking function is defined as the following:

$$score(d,q) = \lambda \times \frac{2 \times overlap(d,q)}{area(d) + area(q)} + (1 - \lambda) \times LuceneScore(d,q)$$

Given a document d and a query q, overlap(d, q) returns the overlap area between d and q, area(d) and area (q) return the area of d and q respectively. λ is the weighting value, and the LuceneScore(d,q) function returns the Lucene ranking score between d and q.

The training data set was used to find the best λ value. Each possible value from 0 to 1 was checked with an increment step of 0.05, and finally $\lambda = 0.90$ was selected for our experiments.

Logistic Regression (LR): This algorithm is developed based on the probabilistic IR model, in which the ranking score for a document to a query is modelled as

$$score(d,q) = -5.68999 + 22.3253 * LuceneScore(d,q) + 2.4842 * \frac{overlap(d,q)}{area(q)} + 1.9533 * \frac{overlap(d,q)}{area(d)}$$

where the coefficients were selected using regression analysis on the training data set.

4.3. Experiment 1: GP Evolution Control Parameters

Before we run our GP algorithm, it is necessary to find the appropriate values of the probability of each GP operator (i.e. p_c for creation, p_o for crossover and p_r for reproduction). Our first experiment conducted was a parameter tuning procedure to find the optimal combination of p_c, p_o and p_r values. The fitness function used in this experiment is the *Fitness2* discussed above, which calculates the fitness value as interpolated precision value at 50% recall. Four different configurations were tested and compared. Table 2 illustrates the p_c, p_o and p_r values of each configuration, as well as the best

	p_o (%)	p_c (%)	p_r (%)	Best Fitness Value
#1	90	10	0	0.766
#2	70	20	10	0.745
#3	50	30	20	0.780
#4	30	50	20	0.769

Table 2: Results of the best fitness value under four different combinations of p_c, p_o and p_r

Recall	Lucene	LC	LR	f1	f2	f3	f4
0.0	0.5647	0.5343	0.5553	0.5017	0.9272	0.7314	0.9323
0.1	0.4302	0.4210	0.4692	0.3788	0.8989	0.7259	0.9130
0.2	0.2913	0.3115	0.3715	0.2771	0.8748	0.7135	0.8755
0.3	0.2269	0.2694	0.3320	0.2531	0.8396	0.7006	0.8513
0.4	0.1675	0.2033	0.2653	0.2444	0.8185	0.6972	0.8348
0.5	0.1471	0.1703	0.2392	0.2353	0.8038	0.6945	0.8239
0.6	0.1353	0.1514	0.2305	0.2206	0.7859	0.6891	0.8066
0.7	0.1297	0.1393	0.2149	0.2020	0.7551	0.6847	0.7709
0.8	0.1234	0.1273	0.2025	0.1962	0.7345	0.6834	0.7486
0.9	0.1124	0.1124	0.1809	0.1865	0.7041	0.6793	0.7176
1.0	0.1106	0.1110	0.1758	0.1830	0.6980	0.6779	0.6981
MAP	0.2014	0.2114	0.2691	0.2365	0.7837	0.6548	0.7973

Table 3: Comparison of the results obtained using four fitness functions with three baselines

fitness values achieved with these configurations. Table 2 indicates that the third configuration (p_o = 50%, p_c =30% and p_r =20%) outperforms others. This configuration was used in the second experiment and remained fixed during the runs.

4.4. Experiment 2: Ranking Function Learning

In this experiment, we run our GP algorithm to learn ranking functions using evolution parameters selected in the experiment 1. The mean interpolated precision values at the standard 11 recall levels of all 100 queries were used as the evaluation matrix. The ranking functions learned using fitness function *Fitness1*, *Fitness2*, *Fitness3* and *Fitness4* are denoted as *f1*, *f2*, *f3* and *f4* respectively. For *Fitness4*, the scaling factor *m* was set to 10. Retrieval performance comparison for the three baseline methods and the ranking functions learned using our method is listed in Table 3. The recall-precision curves are also shown in Figure 5.

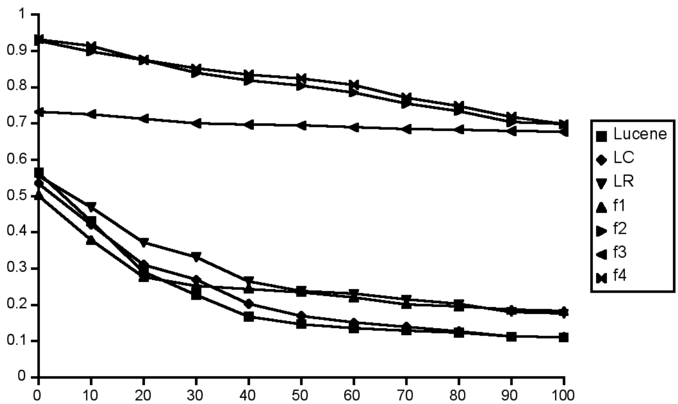


Figure 5: Interpolated precision values at the standard 11 recall levels

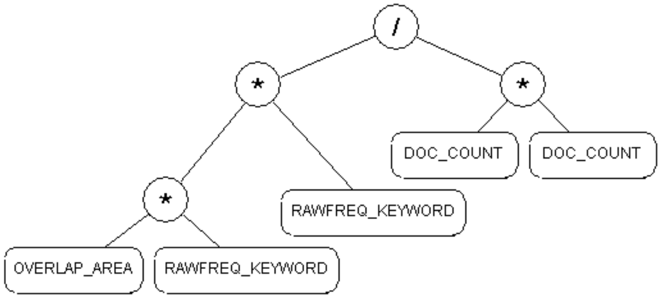


Figure 6: The tree representation of the ranking function learned using Fitness4

As can be seen from Table 3 and Figure 5, the best one of the three baselines was the LR method. Three out of the four GP ranking functions outperformed LR and the performance improvement in the measure of Mean Average Precision (MAP) was from 143.33% to 196.28%. The greatest improvement was obtained with the ranking function *f4* learned using *Fitness4*. The only GP ranking function that didn't show improvement was the *f1* using *Fitness1*. One possible reason for this may be that this fitness function doesn't take into account the order of the retrieved documents. This confirms the hypothesis that the effectiveness of GP algorithms is sensitive to fitness function used, and a careful design and select of fitness functions has a beneficial impact on GP performance.

Figure 6 illustrates the tree representation of the ranking function that achieved the best result. Terminals in the ranking function include both statistics derived from textual content (e.g. *RAWFREQ_KEYWORD*) statistics derived from geographic properties (e.g. *OVERLAP_AREA*). It is important to note that because of the random nature inherent in GP, the mathematical interpretation of the ranking function is usually incomprehensible to humans.

Another interesting comparison is the computation time of GP learning, since GP learning, as well as other evolutionary computation technologies, is a time-consuming procedure. Figure 7 shows the generation numbers of the best fitness value was found for each fitness function. This figure shows that the best fitness value was found after at most 15 generations. Given the computation time is around 10 minutes per generation in our experiment environment, the runtime performance is acceptable.

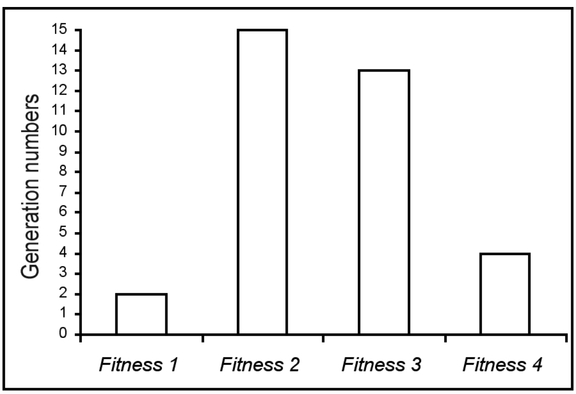


Figure 7: The generation number of the best fitness value found for each fitness function

4.5. Discussion

There are several important observations emerging from this study. Firstly, the experimental results show that ranking functions learned using our GP algorithm can achieve significant improvements on retrieval performance for GIR with acceptable runtime performance. The GP terminals and functions used in our algorithms are very simple and straightforward, but as can be seen from comparisons with baseline methods, very promising results have been obtained with appropriate GP implementations (e.g. evolution strategies and fitness functions).

Secondly, both thematic and geographic similarity measures are critical for GIR ranking. Among all seven ranking methods implemented in our experiments, the Lucene search engine is the only one that totally ignores geographic similarity measures. As can be seen from experimental results, the retrieval performance of Lucene is the worst one among all methods. Although conventional IR has been proven extremely valuable for resolving many problems of information searching and retrieval, it must be complemented by geographic knowledge to produce feasible solutions for GIR applications.

Finally, ranking order must be taken into account when designing GP fitness functions. Previous work has shown that an order-based fitness function can achieve higher performance in various IR tasks. Our experiments confirmed this point by quantifying the performance of fitness functions that use or not use ranking order information.

These observations are important not only for effectively designing GIR ranking mechanisms using GP and other evolutionary computation techniques, but also important for the understanding of the fundamental difference between textual IR systems and GIR systems in how documents are modelled, searched and ranked.

5. CONCLUSIONS AND FUTURE WORK

Ranking of retrieved results plays a very important role in all IR applications. In contrast to conventional IR, GIR ranking must take into account both thematic and geographic similarity measures. This paper described a GIR ranking function learning method where Genetic Programming is used. Selected statistics derived from textual content and geographic properties of documents are used in the learning process. Two experiments have been conducted to evaluate our method. The first one was devoted to the selection of GP evolution strategy, and the second one learned ranking functions using the selected evolution strategy and four different fitness functions. Retrieval performance of the ranking functions learned was compared with other existing ranking mechanisms. The results showed that our method produced significant improvement.

There are several future research directions that we plan to pursue. Firstly, it would be interesting to evaluate our method with bigger document collections in size and higher complexity queries, as it would be useful for the validity of our method. Secondly, our method can be extended by adding other geographic and non-geographic similarity measures as GP terminals. Examples of the former include geographic distances and geographic hierarchical structure. Examples of the latter include population number and economic importance of geographic entities. The addition of these measures may require the support of comprehensive geographic gazetteers and geographic knowledge bases. Finally, there are many other GIR ranking algorithms that have been proposed. It would be a natural further research direction to compare our method with them.

REFERENCES

BAEZA-YATES, R. and RIBEIRO-NETO, B. (1999): *Modern Information Retrieval*, Addison Wesley, 544.

BARTELL, B.T. (1994): Optimizing ranking functions: a connectionist approach to adaptive information retrieval, PhD thesis, University of California. 268.

BARTELL, B.T., COTTRELL, G.W. and BELEW, R.K. (1994): Automatic combination of multiple ranked retrieval systems, in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin. Ireland, 3-6 July, 173-181.

BEARD, K.M. and SHARMA, V.M. (1997): Multidimensional ranking for data in digital spatial libraries, *International Journal on Digital Libraries*, 1(2): 153-160.

CAI, G. (2002): GeoVSM: An integrated retrieval model for geographic information, in *Proceedings of the Second International Conference on Geographic Information Science*, Boulder, CO, United States, 25-28 September, 65-79.

COLEMAN, D.E., JOKIEL, P., HILL, E. and BENTIVOGLIO, N. (2002): *Better Document Management Through Georeferencing*, viewed 24 April 2006 <http://www.esri.com/news/arcuser/0402/docmanage1of3.html>.

DE JONG, K.A. (1975): An analysis of the behavior of a class of genetic adaptive systems, PhD thesis, University of Michigan. 266.

EGGERMONT, J. and VAN HEMERT, J.I. (2001): Adaptive genetic programming applied to new and existing simple regression problems, *Lecture Notes in Computer Science*, 2038: 23-35.

FAN, W.P., GORDON, M.D.P. and PATHAK, P.P. (2005): Genetic programming-based discovery of ranking functions for effective web search, *Journal of Management Information Systems*, 21(4): 37-56.

FOX, E.A. and SHAW, J.A. (1994): Combination of multiple searches, in *Proceedings of the Second Text Retrieval Conference (TREC-2)*, Gaithersburg, Maryland,United States, 31 August -2 September 1993, 243-252.

GEY, F.C. (1994): Inferring probability of relevance using the method of logistic regression, in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin. Ireland, 3-6 July, 222-231.

HILL, L.L. (1990): Access to geographic concepts in online bibliographic files: effectiveness of current practices and the potential of a graphic interface, PhD thesis, University of Pittsburgh. 212.

JEFFERSON, D. (1990): Evolution as a theme in artificial life: The Genesys/tracker system, Perseus Books, 854.

JONES, C.B., PURVES, R., RUAS, A., SANDERSON, M., SESTER, M., KREVELD, M.V. and WEIBEL, R. (2002): Spatial information retrieval and geographical ontologies – An overview of the SPIRIT Project, in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 11-15 August, 387-388.

KOZA, J.R. (1992): *Genetic Programming: On the Programming of Computers by Means of Natural Selection. Complex Adaptive Systems*, MIT Press, 840.

LARSON, R.R. (1996): Geographic information retrieval and spatial browsing, in SMITH, L.C. and GLUCK M. (eds), *Geographic Information Systems and Libraries: Patrons, Maps, and Spatial Information*, Graduate School of University of Illinois 81-124.

LARSON, R.R. and FRONTIERA, P. (2004): Spatial ranking methods for geographic information retrieval (GIR) in Digital Libraries, *Lecture Notes in Computer Science*, 3232: 45-56.

LEE, J.H. (1997): Analyses of multiple evidence combination, in *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia, Pennsylvania, United States, 27-31 July, 267-276.

SALTON, G. (1971): *The SMART retrieval system: experiments in automatic document processing*, Prentice-Hall, 556.

SALTON, G. and BUCKLEY, C. (1988): Term-weighting approaches in automatic retrieval, *Information Processing and Management*, 24(5): 513–523.

TOMLIN, C.D. (1990): *Geographic information systems and cartographic modeling*, Prentice Hall College Div, 249.

TROTMAN, A.S. (2005): Learning to Rank, *Information Retrieval*, 8(3): 359-381.

VOGT, C.C. and COTTRELL, G.W. (1998): Predicting the performance of linearly combined IR systems, in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 190-196.

WOODRUFF, A.G. and PLAUNT, C. (1994): GIPSY: Automated geographic indexing of text documents, *Journal of the American Society of Information Science*, 45(9): 645-655.

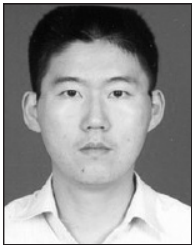
BIOGRAPHICAL NOTES

Linlin Ge is currently an Associate Professor of remote sensing and GPS in the School of Surveying and Spatial Information Systems at the University of New South Wales (UNSW) and a Project Leader and Senior Research Fellow with the Cooperative Research Centre for Spatial Information. He received his BEng (1st Hons) in Optical Engineering from the Wuhan Technical University of Surveying and Mapping (1985), MSc in Crustal Deformation from the Institute of Seismology (1988), and PhD in GPS and remote sensing from the UNSW (2001).

Linlin Ge

Linlin is a Representative on the National Committee of the Remote Sensing and Photogrammetry Commission and foundation member of the Spatial Sciences Institute. He is also the Co-chair of Sub-Commission 4.4 "Applications of Satellite and Airborne Imaging Systems" of the International Association of Geodesy (IAG).

You-Heng Hu is currently a PhD candidate in the School of Surveying and Spatial Information Systems at The University of New South Wales (UNSW). His research interests include Geographic Information Retrieval, Location-based Services, Geographic Information Sciences and Telegeoinformatics, with current projects focusing on the development, evaluation and application of Geographic Information Retrieval systems. You-Heng is a recipient of the Faculty of Engineering Research Scholarship. He received his BEng in Information Control and Engineering from Xi'an Jiaotong University, P.R. China (1995) and Grad.Dip in Computing and Information Technology from The University of New South Wales with the specialization in Database Systems (2004).



You-Heng Hu