Search as if You were in Your Home Town: Geographic Search by Regional Context and Dynamic Feature-space Selection

Makoto P. Kato
JSPS Research Fellow and
Kyoto University, Kyoto, Japan
kato@dl.kuis.kyoto-u.ac.jp

Ohshima Hiroaki
Kyoto University, Kyoto, Japan
ohshima@dl.kuis.kyoto-u.ac.jp

Satoshi Oyama
Hokkaido University, Hokkaido, Japan
oyama@ist.hokudai.ac.jp

Katsumi Tanaka
Kyoto University, Kyoto, Japan
tanaka@dl.kuis.kyoto-u.ac.jp

ABSTRACT

We propose a query-by-example geographic object search method for users that do not know well about the place they are in. Geographic objects, such as restaurants, are often retrieved using an attribute-based or keyword query. These queries, however, are difficult to use for users that have little knowledge on the place where they want to search. The proposed query-by-example method allows users to query by selecting examples in familiar places for retrieving objects in unfamiliar places. One of the challenges is to predict an effective distance metric, which varies for individuals. Another challenge is to calculate the distance between objects in heterogeneous domains considering the feature gap between them, for example, restaurants in Japan and China. Our proposed method is used to robustly estimate the distance metric by amplifying the difference between selected and non-selected examples. By using the distance metric, each object in a familiar domain is evenly assigned to one in an unfamiliar domain to eliminate the difference between those domains. We developed a restaurant search using data obtained from a Japanese restaurant Web guide to evaluate our method.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms

Algorithms, Experimentation

Keywords

Geographic search, query-by-example, dynamic feature-space, heterogeneous domains

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

1. INTRODUCTION

A large amount of geographic object data, such as shops, hotels and landmarks, are available on the Web. Geographic object search has recently received much attention from several Web services. GourNavi[1], which is a Japanese restaurant Web guide, stores over 500,000 restaurants, and Booking.com[2] is an online hotel reservation site with more than 70,000 hotels listed. Keyword and attribute-based search, which require users to translate their search intentions into concrete words or values, are often used in those commercial sites. However, it is difficult to explain such an object by using keywords or attributes, especially for a visitor in a place he/she does not know well. For example, if you visited Japan, you would find that almost all of the restaurants are different from those in your home town. In that case, you might find it difficult to make a query without knowledge on what kinds of food are popular and what the average cost is, even though you could query easily in your home town.

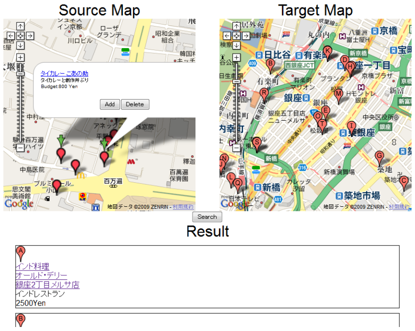
Thus, we adapt a query-by-example paradigm for geographic object search, which has been used in multimedia retrieval (often called *content-based retrieval*) [3]. The advantage of the query-by-example paradigm is that users do not need to express their search intentions explicitly, but only choose examples they think to be relevant in a well known domain. Even if you do not have any knowledge about a place where you wants to find information, you can *search as if you were in your home town* by imagining what objects would be relevant there.

Figure 1 shows the query-by-example interface we developed for geographic object search. The interface presents two maps: one is a known place to the user, e.g., his/her hometown, and the other is an unknown place containing objects to be retrieved. They are called *source map* and *target map*, respectively. Users can select which area is shown for both maps, and which objects in the source map are relevant to their search intentions. Then, the closest objects to an input are returned as search results.

To rank objects based on an input query, the distance between objects should be defined. (Note that a term *distance* basically means distance in a feature-space in this paper. Only a phrase *geographic distance* means distance in the real world.) There are mainly two problems in measuring distance.

First, the notion of distance (or similarity) depends on users. For example, the distance between two restaurants, one French for a

[1] http://www.gnavi.co.jp/
[2] http://www.booking.com/



タイカレー こあの助
タイカレーと創作丼ぶり
Budget:800 Yen

A
インド料理
オールド・デリー
銀座2丁目メルサ店
インドレストラン
2500Yen

B

Figure 1: Query-by-example interface for geographic object search.

$40-meal, and the other Japanese for a *$40*-meal, can be either close or far by focusing on the prices or styles. Attributes that are focused on vary by user. Thus, the adaptive distance metric should be considered, which gives weight to important dimensions in the distance calculation.

The second problem is measuring distance between objects in heterogeneous domains, i.e., a familiar and an unfamiliar place. Consider the distance in a feature-space between restaurants in Japan and China. There is a large gap between price range, popular food, styles, and scales of geographic distance. The differences raise the following questions: *1,000 yen* and *70 renminbi*, how much are they different, and whether *5 km* in Japan and *5 km* in China are sensuously the same. Therefore, a method to deal with these heterogeneous dimensions is required for the proposed geographic search.

We followed the adaptive distance model proposed by Ishikawa et al. (MindReader) [2], and improved the model by amplifying the difference between selected and non-selected examples. For the second problem, we modeled it as an assignment problem between heterogeneous domains. Each object in one domain is mapped to one in another to minimize the sum of all the distances between the assigned pairs. The assignment bridges the gap between heterogeneous domains, and makes it possible to find correspondence to given examples without directly calculating their distance. We evaluated the effectiveness of our proposed method by comparing it with a baseline.

2. MODEL

Geographic objects have several attributes, including position, and each attribute value is represented as a point in n-dimensional space. An object is defined as $\mathbf{o} = (\mathbf{o}_{a_1}, \mathbf{o}_{a_2}, \ldots, \mathbf{o}_{a_M}, \mathbf{o}_{\mathrm{pos}})$, where a vector $\mathbf{o}_{a_i}$ is an attribute value for an attribute a_i such as budget for restaurants. The vector $\mathbf{o}_{a_i}$ can represent a scalar, such as budget for restaurants, or descriptions as a term frequency-inverse document frequency (tf-idf) vector. The vector $\mathbf{o}_{\mathrm{pos}}$ is a position vector containing latitude and longitude.

Query-by-example for geographic object search enables users to select examples in a familiar place, and retrieves and ranks objects in an unfamiliar place based on their distances. Users can also choose a familiar place and an unfamiliar one, which are called *source domain* $\boldsymbol{O}_s \subset \boldsymbol{O}$, and *target domain* $\boldsymbol{O}_t \subset \boldsymbol{O}$ ($\boldsymbol{O}$ is a set of all the objects). We assume that a user knows all the objects in the source domain $\boldsymbol{O}_s$. Since $\boldsymbol{O}_s$ can be changed by individual users, it is a reasonable assumption in this model.

From the source domain $\boldsymbol{O}_s$, users can select a subset as a query to search for objects in the target domain $\boldsymbol{O}_t$. Consequently, a set of queries $\boldsymbol{Q}$, and data $\boldsymbol{D}$ to be retrieved in query-by-example for geographic object search are defined as follows: $\boldsymbol{Q} = 2^{\boldsymbol{O}_s}$, $\boldsymbol{D} = \boldsymbol{O}_t$. Given a query $\boldsymbol{Q}_i \in \boldsymbol{Q}$, objects in $\boldsymbol{D}$ are ranked using a ranking function $\mathrm{Rank}(\boldsymbol{Q}_i, \mathbf{d}_j) = f(\boldsymbol{Q}_i, \mathbf{d}_j, \mathrm{dist}(\cdot,\cdot))$. The term $\mathrm{dist}(\cdot,\cdot)$ is the distance between selected objects and an object, i.e., $\mathrm{dist} : \boldsymbol{Q} \times \boldsymbol{O} \rightarrow \mathbb{R}$. We explain a method of determining a distance metric based on dynamic feature-space selection in Section 3. The details of $\mathrm{Rank}(\boldsymbol{Q}_i, \mathbf{d}_j)$ are discussed in Section 4.

3. DYNAMIC FEATURE-SPACE SELECTION

This section presents the variable distance function proposed in MindReader, explains the difference from our problem, and proposes an adaptation to geographic object search.

3.1 Formulation of MindReader

In MindReader, the distance function between a query $\boldsymbol{Q}_i$ and an object $\mathbf{o}$ is $\mathrm{dist}(\boldsymbol{Q}_i, \mathbf{o}) = (\mathbf{o} - \mathbf{m})^T \mathbf{W} (\mathbf{o} - \mathbf{m})$ where it is assumed that a user has an *ideal* query $\mathbf{m}$ and an expected distance corresponding to a symmetric matrix $\mathbf{W}$ in mind, and the problem is to predict $\mathbf{m}$ and $\mathbf{W}$ from the given set of examples $\boldsymbol{Q}_i$. The basic idea of the prediction is to minimize the distance between selected examples $\boldsymbol{Q}_i$ and the ideal query $\mathbf{m}$. If a user had an ideal query and an ideal distance, examples selected by the user would be close to the ideal query based on his/her distance metric. According to the assumption, the prediction of the ideal query $\mathbf{m}$ and the matrix of the ideal distance matrix $\mathbf{W}$ is formulated as a minimization problem: $\min_{\mathbf{m},\mathbf{W}} \sum_{\mathbf{q}_k \in \boldsymbol{Q}_i} g_k (\mathbf{q}_k - \mathbf{m})^T \mathbf{W} (\mathbf{q}_k - \mathbf{m})$ subject to the constrain $\det(\mathbf{W}) = 1$. The scalar g_k is a goodness value for selected examples, and the default value is 1 (it can be multi-level.)

The problem was solved analytically. The ideal vector $\mathbf{m}$ equals to an average of selected examples, and the matrix of the ideal distance is proportional to an inverse covariance matrix.

3.2 Dynamic Feature-space Selection by Difference Amplification

The problem we raised in Section 2 is different from that of MindReader. There are selected, and also non-selected examples $\bar{\boldsymbol{Q}}_i = \boldsymbol{O}_s - \boldsymbol{Q}_i$ in our problem. Both types of examples contain meaningful information, and considering the non-selected examples makes it possible to see the **regional contexts**. We could imagine the difference of the meanings between an inexpensive restaurant in a place where there are many expensive ones and the same one in a place where there are many inexpensive ones. The former strongly indicates that price is very important, and the latter does not.

The selected examples can be regarded as positive ones, however, the non-selected examples are not always negative. This is why we cannot use the method of utilizing both positive and negative examples proposed by Ashwin et al. [1].

Therefore, we propose a method for predicting the distance metric by difference amplification. When a user chooses examples as positive ones, it is assumed that he/she should select the closest ones to the ideal query $\mathbf{m}$, while the others are not selected because he/she considers them far from the ideal query $\mathbf{m}$. The basic idea of our approach is to **minimize the distance between selected examples $\boldsymbol{Q}_i$ and the ideal query $\mathbf{m}$, and also maximize the distance between non-selected examples $\bar{\boldsymbol{Q}}_i$ and the ideal query $\mathbf{m}$**. The non-selected examples do not affect the ideal query $\mathbf{m}$. With the ideal query $\mathbf{m}$ fixed to the average of $\boldsymbol{Q}_i$, non-selected examples change the ideal distance matrix $\mathbf{W}$ to amplify the difference between selected and non-selected examples.

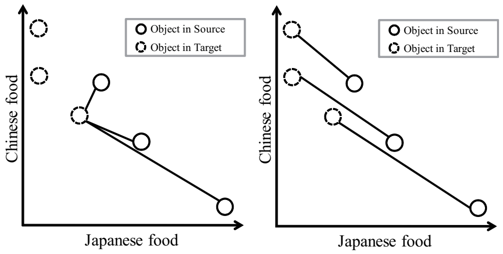


Figure 2: Problem of ranking based on distance in heterogeneous domains. Horizontal axis represents how Japanese-like food restaurant serves. Vertical axis represents the same for Chinese-like food.

The modified minimization problem of MindReader is:

$$\min_{\mathbf{W}} \frac{1}{|\boldsymbol{Q}_i|}\sum_{\mathbf{q}_k\in\boldsymbol{Q}_i} g_k(\mathbf{q}_k-\mathbf{m})^T\mathbf{W}(\mathbf{q}_k-\mathbf{m}) - \frac{\alpha}{|\bar{\boldsymbol{Q}}_i|}\sum_{\mathbf{o}\in\bar{\boldsymbol{Q}}_i} h(\boldsymbol{Q}_i,\mathbf{o})(\mathbf{o}-\mathbf{m})^T\mathbf{W}(\mathbf{o}-\mathbf{m}) + \frac{\rho}{2}\|\mathbf{W}-\hat{\mathbf{W}}\|^2, \qquad (1)$$

subject to the constraint: $\|\mathbf{W}\| = 1$, $w_{ij} \geq 0$ where w_{ij} is an i-j element of matrix $\mathbf{W}$, and $\mathbf{m}$ is an average of $\boldsymbol{Q}_i$ (weighted by g_k,) i.e., $\mathbf{m} = \frac{1}{N_g}\sum_{\mathbf{q}_k\in\boldsymbol{Q}_i} g_k\mathbf{q}_k$.

The major modifications can be seen in the range of summation, the ideal vector, and the penalty term. First, we take a sum for all the examples, including non-selected examples with negative scalar values $-h(\boldsymbol{Q}_i, \mathbf{o})$. The function $h(\boldsymbol{Q}_i, \mathbf{o})$ has some variations, which are explained in our experiment. Second, the ideal vector is fixed to the average of selected examples, which is no longer a variable in the minimization, but the same as the solution of the ideal query $\mathbf{m}$ in MindReader. Finally, in judging the distance between objects, some attributes are commonly important, whereas others are meaningless. Thus, we pre-define a standard distance $\hat{\mathbf{W}}$, and an extraordinary distance obtained with the prediction is given a penalty $\|\mathbf{W} - \hat{\mathbf{W}}\|^2$.

4. RANKING METHOD BY BRIDGING HETEROGENEOUS DOMAINS

The simplest method for ranking objects in a target domain is using the distance metric as a ranking function: $\text{Rank}(\boldsymbol{Q}_i, \mathbf{d}_j) = -\text{dist}(\boldsymbol{Q}_i, \mathbf{d}_j)$, which indicates that closer objects to a given query would receive higher ranks. This definition seems to be reasonable, however, as mentioned in Section 1, there is a fundamental problem concerning the distance between objects in heterogeneous domains. The left side of Figure 2 shows a simple example of this problem. There are many Japanese restaurants in Japan, and a few Japanese and many Chinese restaurants in China. If a source domain was set to Japan, and a target domain to China, whatever a user selects from the source domain, only the Japanese restaurant in China would be returned at the top of search results. Even if the most Chinese-like restaurant in Japan was selected, the search result would not change.

Therefore, we propose a method for eliminating the gap between heterogeneous domains by solving an assignment problem between objects in the two domains. As seen in the right side of Figure 2, the best assignment **minimizes the sum of the distances between the pairs, where all the objects must be evenly assigned to one or more objects.** The even assignment matches one distribution with another, and makes results different for different queries considering the relativeness of features. It can be easily interpreted as an assignment problem in a bipartite graph.

The bipartite graph of the source-target domains is a pair of nodes and edges, $B = (V, E)$, where $V = \boldsymbol{O}_s \cup \boldsymbol{O}_t$, $E \subset \boldsymbol{O}_s \times \boldsymbol{O}_t$. The distance function d between two objects is defined as $d(\mathbf{o}_i, \mathbf{o}_j) = (\mathbf{o}_i - \mathbf{o}_j)^T\mathbf{W}^*(\mathbf{o}_i - \mathbf{o}_j)$, where $\mathbf{W}^*$ is the optimal distance matrix obtained in Section 3.

The variable $x_{i,j}$ is defined as 1 if $(\mathbf{o}_i, \mathbf{o}_j) \in E$, otherwise 0. Assume, without loss of generality, that the size of $\boldsymbol{O}_s$ is less than that of $\boldsymbol{O}_t$. The assignment problem leads to the minimization problem for edges E as follows.

$$E^* = \operatorname*{argmin}_E \sum_{\mathbf{o}_i\in\boldsymbol{O}_s}\sum_{\mathbf{o}_j\in\boldsymbol{O}_t} d(\mathbf{o}_i, \mathbf{o}_j)x_{i,j}, \qquad (2)$$

subjects to $\left\lfloor \frac{|\boldsymbol{O}_t|}{|\boldsymbol{O}_s|} \right\rfloor \leq \sum_{\mathbf{o}_i\in\boldsymbol{O}_s} x_{i,j} \leq \left\lceil \frac{|\boldsymbol{O}_t|}{|\boldsymbol{O}_s|} \right\rceil$ and $\sum_{\mathbf{o}_j\in\boldsymbol{O}_t} x_{i,j} = 1$ where $\lfloor x \rfloor = \max\{n | n \in \mathbb{Z} \wedge n \leq x\}$, and $\lceil x \rceil = \min\{n | n \in \mathbb{Z} \wedge x \leq n\}$. The first restriction on the variable $x_{i,j}$ forces objects in a source domain to have almost an equal amount of edges to those in a target domain. The second restriction makes objects in a target domain have an edge to ones in a source domain.

By using the optimal assignment E^*, $\text{Rank}(\boldsymbol{Q}_i, \mathbf{d}_j)$ takes the average of the distance between selected examples $\boldsymbol{Q}_i$ and objects assigned to the data $\mathbf{d}_j$. The ranking function is defined as follows: $\text{Rank}(\boldsymbol{Q}_i, \mathbf{d}_j) = -\frac{1}{N_{\mathbf{d}_j}}\sum_{(\mathbf{o}_s,\mathbf{d}_j)\in E^*} \text{dist}(\boldsymbol{Q}_i, \mathbf{o}_s)$ where $N_{\mathbf{d}_j} = |\{\mathbf{o}_s | (\mathbf{o}_s, \mathbf{d}_j) \in E^*\}|$.

5. EXPERIMENT

This section first describes our implementation of our query-by-example method for geographic object search. We then present the set up of a test set for our experiment and discuss the experimental results.

5.1 Implementation

The geographic objects we used were obtained from GourNavi, which is a service for introducing Japanese restaurants. A map interface was implemented with Google Maps API[3] as can be seen in Figure 1, which allows users to select examples in a familiar place to search for restaurants in an unfamiliar place.

Restaurant objects were obtained through the GourNavi Web service[4]. The number of retrieved objects was 46,945, and were stored and indexed by latitude and longitude. There are various types of attributes, text, float, integer, and set; however, some attributes are insignificant for object distance. We only used five attributes for measuring distance between objects; name, category, category label, introduction, and budget.

The standard tf-idf method was used for a text (name, category, and introduction) and set attribute (category label) value. These attribute values are very sparse and are represented as points in a high dimensional space. For example, 27,212 dimensions are allocated for text attribute values, and 158 for category labels (e.g., *Japanese* and *Chinese*). Thus, we applied *latent semantic analysis* to compress their dimensions into 50 for text and 20 for category labels. The budget attribute was also normalized so that the maximum distance is 2, which is the same as the other attributes.

5.2 Experimental Settings

Four volunteers manually created a test set for performance evaluation. The test set consisted of search intentions, queries, and data

[3] http://www.google.com/apis/maps/

[4] http://api.gnavi.co.jp/

Table 1: Search intentions, source maps, and target maps.

Search intentions	
ID	Content
1	Restaurants for around 1,000 yen
2	Restaurants serving spicy food
3	Restaurants serving sea food
4	Expensive restaurants
5	Restaurants serving special local foods at modestly high prices

Source maps		
ID	Area	# of objects
1	Tokyo	55
2	Nagoya	55
3	Osaka	59
4	Sapporo	51

Target maps			
ID	Area	# of objects	Avg of Kappa
1	Kyoto	49	0.94
2	Kobe	50	0.86

Table 2: Averages of nDCG, MAP, and @1.

	nDCG	MAP	@1
MindReader	0.675	0.340	0.375
DA	**0.739**	**0.460**	**0.500**
DA+RA	0.693	0.377	0.325

with a relevance score. The search intentions, source maps (where users select examples as a query,) and target maps (where examples are retrieved) are listed in Table 1. As can be seen, we had 20 queries and 2 data sets to be retrieved. On an average, 3.4 examples were selected as a query. In our experiment, the combinations, i.e., 40 tests, were tried with each method described in the next section.

To validate the effectiveness of our method, MindReader was used for comparison, explained in Section 2 (**MindReader**). Our proposed method for predicting the distance metric by Difference Amplification is represented as **DA**. The Ranking by Assignment between heterogeneous domains is represented by **RA**. DA does not use the method RA, and we compared three methods MindReader, DA, and DA+RA in this experiment. In this experiment, the distance matrices were limited to a diagonal matrix.

The function $h(\boldsymbol{Q}_i, \mathbf{o})$ and parameter α in Equation 1 were determined in a preliminary experiment; $h(\boldsymbol{Q}_i, \mathbf{o}) = 1$ if the distance in a feature-space between the average vector of $\boldsymbol{Q}_i$ and the vector $\mathbf{o}$ is less than β, otherwise 0. Note that the distance is normalized by the average distance between selected objects $\boldsymbol{Q}_i$, and $\alpha = 2.0$ and $\beta = 3.0$. The parameter ρ was fixed to 1.

5.3 Experimental Results

5.3.1 Comparison with Previously Proposed Method

The comparison of our proposed methods with MindReader is shown in Table 2. DA method obtained higher nDCG and MAP scores than MindReader, and there was a significant difference between them ($p < .05$). This was because the small number of given examples made it difficult to predict the distance metric for each search intension. Using the method DA, we were able to use more information including selected examples and also non-selected ones, and estimate the distance metric robustly. On the other hand, even though DA+RA got higher scores than MindReader except @1 precision, it failed to improve the performance from only DA.

5.3.2 Case Study for Ranking Method by Assignment

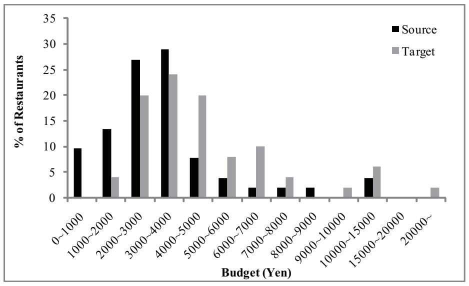


Figure 3: Statistics of case study.

Table 3: Effect of ranking by assignment.

		Style	Budget
Selected		French	7,000Yen
		French	6,000Yen
	Rank	Style	Budget
Top 3 (DA)	1	French & Wine bar	6,000Yen
	2	Casual French	7,000Yen
	3	Italian & Cafe	2,500Yen
Top 3 (DA+RA)	1	French & Wine bar	6,000Yen
	2	Casual French	7,000Yen
	3	Fictive Japanese	10,000Yen

We explain a case study to show the effects by DA+RA. Figure 3 shows, in the source domain of this case, restaurants are relatively inexpensive, on the other hand, in the target domain, some of restaurants are so expensive. There is a gap on the average budget between the source and target domains. Given two expensive French examples as a query, in Table 3, DA+RA returned a little different result from only DA. At the third rank, DA returned a very cheap Italian restaurant, while DA+RA presented rather expensive Japanese one.

The difference was made by their budget distributions. Imagine a user that selected the two restaurants as a query. There are a few restaurants for more than 6,000 Yen in the source domain (around 10%,) and the two restaurants should be considered the most expensive restaurants for the user. On the other hand, in the target domain, there are more restaurants for over 6,000 Yen (around 20%,) and the most expensive restaurants should be for more than 9,000 Yen (around 10%.) Considering the gap between the heterogeneous domains, the result by DA+RA was more reasonable.

6. CONCLUSION

We proposed a method of searching for geographic objects in an unfamiliar place with a query-by-example in a familiar place. Even if a user does not have any knowledge about a place where he/she wants to find information, the proposed method enables he/she to make a query by selecting relevant examples in a well known place.

7. ACKNOWLEDGMENTS

This work was supported in part by the following projects and institutions: Grants-in-Aid for Scientific Research (Nos. 18049041, 21700105 and 21700106) from MEXT of Japan, a Kyoto University GCOE Program entitled "Informatics Education and Research for Knowledge-Circulating Society," and the National Institute of Information and Communications Technology, Japan.

8. REFERENCES

[1] T. V. Ashwin, R. Gupta, and S. Ghosal. Adaptable similarity search using non-relevant information. In *Proc. of VLDB 2002*, pages 47–58, 2002.

[2] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *Proc. of VLDB 1998*, pages 218–227, 1998.

[3] A. Yoshitaka and T. Ichikawa. A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81–93, 1999.