MrCoM: A Cost Model for Range Query Translation in Deep Web Data Integration

Fangjiao Jiang, Linlin Jia
School of Information
Renmin University of China
Beijing, 100872
China
{jiangfj, jialinlin510}@gmail.com

Weiyi Meng
Dept. of Compute Science
SUNY Binhamton
Binhamton, NY 13902
USA
meng@cs.binhamton.edu

Xiaofeng Meng
School of Information
Renmin University of China
Beijing, 100872
China
xfmeng@ruc.edu.cn

ABSTRACT
Due to the autonomy of web databases, a major challenge for query translation in a Deep Web Data Integration System is the lack of cost models at the global level. In this paper, we propose a Multiple-regression Cost Model (MrCoM) based on statistical analysis for global range queries that involve numeric range attributes. Using the MrCoM, the query translation strategy for new global range queries can be inferred. We also propose a Pre-processing-based Stepwise Algorithm (PSA) for selecting significant independent variables into the MrCoM. Experimental results demonstrate that the fitness of the MrCoM is good and the accuracy of the query strategy selection is high.

Categories and Subject Descriptors
H.3.4 [**Information Storage and Retrieval**]: Systems and Software – Distributed Systems; H.3.5: Online Information Services –Web-based Services.

General Terms
Algorithms, Performance, Design, Experimentation.

Keywords
Deep Web Data Integration System, Multiple-regression Cost Model, query translation

1. INTRODUCTION
The Deep Web is a vast and rapidly growing source of information, which is about 500 times larger than the Surface Web [1]. It has become more and more difficult for users to find their interested information. The goal of Deep Web data integration is to provide a unified interface for a specified domain (e.g., airfares) and to automatically retrieve the relevant information from multiple Web databases according to the global query the user issued. The user does not need to know where the data is stored as well as how the result is obtained. Some related work includes Web database crawling [3, 4], interface integration [5, 6] and result extraction [7, 8]. Query translation which is responsible for translating a global query of the integrated interface to a local query of a Web database has not been carried out widely.

Often the query translation strategy is not unique, so choosing the best translation strategy is the task of the global query translator. The crucial challenge for query translation in the Deep Web data integration environment is that some local optimization information, for example, the size of the database, the index or cluster index on some attributes, the profile of the database, may not be known or accurately known by the global query translator due to local autonomy. Traditional query translation techniques may not be suitable because precise optimization information is required. In other words, autonomous Web databases can be viewed as a black box whose optimization information is hidden from the global translator. What we can do is to predict the cost from the external characteristics of the black box.

What we will focus on in this paper is query translation of range selection queries on some range attributes with numeric data type. As Figure 1 shows, the "Price" attribute is an attribute of this type. Consider a global query that contains an original range {Price, (8000, 12000)}, it can be translated into two local queries, one of which contains a sub-target range {Price Range, (5000, 10000)} and the other contains a sub-target range {Price Range, (10000, 15000)}. The target range {Price Range, (5000, 15000)} is the union of these sub-target ranges. And then filter out the undesired results whose prices are between 5000 and 8000 and between 12000 and 15000. Usually, in order to access all the desired results, the target range must contain the original range, i.e., the coverage. Meanwhile, in order to reduce the undesired intermediate results to the minimum, we ought to choose the minimum target range, i.e., the minimum coverage. In the above example, the target range is exactly a minimum coverage of the original one. But as the price range increases in the global query, the number of local queries will also increase, which will lead to higher cost because more local queries imply more invocations to a local Web database and more requests for network connections. Is it a better strategy that we translate the user query into {Price Range, (any/all)} and then filter out undesired results?

The former is usually adopted currently and it will be called the ***minimum coverage strategy***. Obviously, this method will lead to higher processing cost on the server side because usually more than one local query will be posed to the Web database but lower cost on network transmission and filtering because the intermediate results will contain less undesired information. On the other hand, the latter strategy, which will be called the ***maximum coverage strategy***, needs to pose only one query to the

Web database but the intermediate results will contain more undesired information which may lead to lower processing cost on the server side but higher cost on network transmission and filtering. The question is, given a global query, how to select the more efficient strategy between these two options? This is exactly what we would like to answer in this paper.

In some cases, there is no need to choose a query translation strategy. As we know, when a form is submitted, the web browser may use one of two methods "get" or "post" to send an HTTP request with the parameters and their values to the server. With "get", the parameters are appended to the action and included as part of the URL in the HTTP request (e.g., http://autotrader.co.nz/UsedItemResults.aspx?N=0&sid=&Nf=P_ Price|BTWN+10000+20000). In this case, the query posted to the integrated interface can be precisely translated to a query against the local interface. However with "post", the parameters are sent in the body of the HTTP request, making equivalent translation impossible. Many web databases only support the "post" method. For such Web databases, selecting a more efficient translation strategy is desirable.

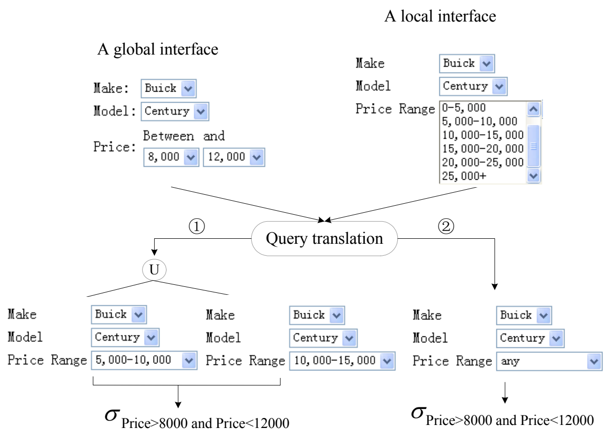

Figure 1. An example of range query translation strategy

The cost of a global range query may be influenced by many factors, for example, the range of the global query. In this paper, we consider these factors as much as possible. By recording and analyzing the costs of the two strategies (i.e., the minimum coverage strategy and the maximum coverage strategy) for a set of sample global queries against the relevant local database, the impact of various factors can be determined and a cost formula can be derived and used to select the translation strategy for any new global range query.

The contributions of this paper include the following:

1. Propose a Multiple-regression Cost Model, MrCoM, which is a statistical-based approach for translating the range query at the global level.

2. Propose a Pre-processing-based Stepwise Algorithm (PSA) for selecting significant independent variables into the MrCoM.

3. Classify the global range queries into three types. For the goodness of fit, we build different models for each of the query types.

4. Validate the effectiveness of the method by experiments.

The rest of the paper is organized as follows. Section 2 reviews some related works. Section 3 gives the definition of the two strategies and presents the classification of the global range query. Section 4 analyzes the types of factors that will affect the cost, proposes the Multiple-regression Cost Model (MrCoM) and introduces the procedure of constructing it. Experimental results are reported in section 5. The paper is concluded in section 6.

2. RELATED WORKS

Query translation at global level has been investigated in Multiple Database System [2] (MDS) for a long time. In spite of lacking the information the optimization requires, such as cardinality of the tables or index information of the schemas, MDS can still make use of the global or local catalog information to perform global query optimization to some extent. One similar work to ours is [2]. In [2], Q. Zhu proposes an optimization cost model in MDS on global level. Due to awareness of the local catalog information, the queries can be classified into some homogeneous classes such as unary query, join query, etc. In each class, the costs of sample queries were recorded for generating the cost model which was used to infer the cost of a new query. Compared to MDS, Deep Web Data Integration System has much less information the optimization requires due to unavailability of the catalog information. Hence, it is much harder to choose the better strategy for query translation on global level in this environment. In addition, we consider many factors that influence the cost which are not mentioned in [2], such as query range, network transmission and data distribution, etc.

The other similar work to ours is [10]. In [10], Z. Zhang proposes an optimized query translation in Deep Web Data Integration System focusing on text type attribute with some constraint. The constraint is restriction to the query condition, for example, *any*, *all* or *exactly*. The minimum search space is computed but the cost of the query is not considered.

In this paper, we are interested in how to select the more efficient query translation strategy for any global query that has range conditions on numeric range attributes. In short, the issue we focus on and the method we propose are different from [2, 10].

3. QUERY TRANSLATION STRATEGY

Definition 3.1. A *range query* is a query involving a set of attribute-value pairs with at least one numeric range attribute.

In this paper, we consider a global range query that contains one numeric range attribute, like {<Make, "Audi">, <Model, "A6">, <Price, "50000-60000">}, here Price is a numeric range attribute and its value is between 50000 and 60000.

For the rest of this paper, the range of a global query refers to the range values of the numeric range attribute.

Definition 3.2. *Minimum Coverage Strategy (MinCS).* Given a global query Q_i in the integrated interface, the local query Q* in a web database interface is a query translation using the minimum coverage strategy if the following conditions are satisfied at the same time:

1. Q* is a valid query, i.e., acceptable to the local query interface.
2. Q* semantically subsumes Q_i, i.e., for any database instance D_j, $Q_i(D_j) \subseteq Q^*(D_j)$. The expression denotes that the retrieved information of Q_i is a subset of that of Q*.
3. There does not exist any query Q** which satisfies 1 and 2 and $Q^{**}(D_j) \subset Q^*(D_j)$.

Condition 2 guarantees that all the information the user needs is retrieved and condition 3 guarantees that the retrieved information is the minimum possible. This strategy minimizes the amount of data to be sent from the Web database and minimizes the effort to filter out undesired results.

But when the original range becomes large, the cost of MinCS will be high because the number of local queries to the Web database will increase. For example, in Figure 1, if a global query contains an original range {price, (8000, 50,000)}, the number of local queries will increase to 9.

Definition 3.3. *Maximum Coverage Strategy (MaxCS).* Given a query Q_i in the integrated interface, the query Q* in a web database interface is a query translation using the maximum coverage strategy if the following conditions are satisfied at the same time:

1. Q* is a valid query, i.e., acceptable to the local query interface.
2. Q* semantically subsumes Q_i, i.e., for any database instance D_j, $Q_i(D_j) \subseteq Q^*(D_j)$. This expression denotes that the result set of Q_i is the subset of that of Q*.
3. There does not exist any query Q** which satisfies 1 and 2 and $Q^{**}(D_j) \supset Q^*(D_j)$.

Usually, a range attribute has value *any* or *all* which means no restriction on this attribute. Under this condition, we can obtain the maximum result set and then filter out the undesired results. The advantage of this strategy is that the query needs to be submitted to the local database only once.

But if the range of a query is relatively small (e.g., 50-100) compared to the full range (e.g., 0-10,000), the results returned by MaxCS will contain a lot of undesired information, which increases the network transmission and filtering cost.

Note that for a given global query, the absolute cost of a query translation strategy (for example, using MinCS) may be quite different with respect to different web databases in the same domain due to many factors, e.g. the database size, etc. Moreover, our goal is to choose a proper strategy from the two strategies to lower the cost, not just to know the cost of MinCS or MaxCS. To derive a cost estimation formula for one domain, the relative cost between MinCS and MaxCS is computed as follows.

Definition 3.4. *Relative Cost (RC)* is the ratio of the cost of MinCS to that of MaxCS.

$$\text{Relative Cost(RC)} = \frac{the \quad \cos t \quad of \quad MinCS}{the \quad \cos t \quad of \quad MaxCS} \qquad (1)$$

If the cost of MinCS is higher than that of MaxCS, i.e., if RC is larger than 1, we will select MaxCS; otherwise we select MinCS.

4. MrCoM for Range Query Translation

In this section, we will construct the Multiple-regression Cost Model, MrCoM, for range query translation based on the analysis of the types of factors and the classification of the global range query.

4.1 Multiple Regression Model

Multiple regression represents a statistical relationship between the dependent variable and the independent variables. In our application, the dependent variable is the *Relative Cost*. The independent variables are the factors that affect the *Relative Cost*. Let X_1, X_2, …, X_k be k independent variables. If the dependent variable Y tends to vary in a linear manner with respect to the independent variables X's, a multiple regression model [11] is defined as:

$$Y = A_0 + A_1X_1 + A_2X_2 + \cdots + A_kX_k + \varepsilon \qquad (2)$$

where A_0, A_1, …, A_k are regression coefficients, X_1, X_2, …, X_k are the values of the independent variables, and ε denotes the random error term. A_0, A_1, …, A_k are unknown constants and X_1, X_2, …, X_k are known values from sample experiments.

To evaluate the fitness of the multiple regression model, some test statistics are usually estimated, e.g., standard error δ^2 and coefficient of determination R^2.

$$\delta^2 = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 /(n-(k+1)) \qquad (3)$$

$$R^2 = 1 - \sum_{i=1}^{n}(Y_i - (\sum_{j=1}^{n}Y_j)/n)^2 \qquad (4)$$

δ^2 is an indication of the accuracy of estimation which is obtained by the multiple regression formula. Y_i is an observed value. $\hat{Y}_i$ is the corresponding fitted value. The smaller the δ^2 is, the more successful the regression model is. R^2 is the proportion of variability in the dependent variable Y explained by the independent variable X_i. The closer to 1 the value of R^2 is, the more successful the regression model is.

4.2 The Types of Factors

For the two different strategies, the total cost can both be divided into three stages: the first stage is to retrieve the tuples that satisfy the local query condition from the local database, the second stage is to transmit the retrieved tuples from the local database to the integrated system, and the third stage is to filter out the undesired tuples to obtain those that satisfy the global query condition. The total cost can be expressed by the formula as follows:

$$T_{total} = T_r + T_t + T_f \qquad (5)$$

where T_{total} represents the total cost, T_r represents the time needed to retrieve the tuples, T_t represents the time needed to transmit the retrieved tuples and T_f represents the time needed to filter out the undesired tuples.

In following formulas, $T_{total(min)}$ and $T_{total(max)}$ represent the total cost time of *MinCS* and *MaxCS* respectively.

$$T_{total(\min)} = T_{r(\min)} + T_{t(\min)} + T_{f(\min)} \quad (6)$$

$$T_{total(\max)} = T_{r(\max)} + T_{t(\max)} + T_{f(\max)} \quad (7)$$

Given a specific web database, we assume that the values of the attributes except the range attribute are fixed. Intuitively, for any global range query, $T_{total(max)}$ will not change because $T_{r(max)}$, $T_{t(max)}$ and $T_{f(max)}$ are independent of the query. On the contrary, $T_{total(min)}$ may change greatly because $T_{r(min)}$, $T_{t(min)}$ and $T_{f(min)}$ will change when some factors (e.g., the range of the global query) are different. Therefore the *Relative Cost* will be different as some factors change.

According to our observations, when translating a global range query to a web database, the following types of factors (i.e., F1, F2,…) will likely affect the *Relative Cost*, which will directly influence strategy selection i.e., MinCS or MaxCS.

F1: The range of a global query. The larger the range of a global query is, the higher the Relative Cost is. This is because when using MinCS, the number of local queries submitted to the local database will likely increase and the retrieving cost on the server side will increase consequently. Moreover, the transmission cost and filtering cost will also increase with the return of more results. On the other hand, the cost of using MaxCS remains the same.

F2: The full range of a local attribute. The larger the full range of a local attribute is, the lower the Relative Cost is. This is because that the range of a global query may be much smaller than the full range. When using MinCS, the retrieving cost, transmission cost and filtering cost would be lower relative to using MaxCS.

F3: The granularity of a local attribute range. The granularity of a local attribute range is the width of each of the attribute range value. For example, in Figure 1, the granularity of the price range is 5,000. The smaller the granularity of a local attribute range is, the higher the Relative Cost is. This is because with smaller granularity and when using MinCS, the number of local queries submitted to the local database will increase and the retrieving cost on the server side would increase accordingly. On the other hand, the cost of using MaxCS does not change.

F4: The distribution of the attribute value or the position of the range. The distribution of the attribute value is denoted by the number of tuples in different ranges of the attribute. Usually the distribution of the values is not uniform. But regular distribution pattern may exist for some attribute in one domain. For example, based on our observation of 30 car Web sites, the number of the cars with different prices approximately follows the normal distribution. So the position of the range which is denoted by the average of the maximum and the minimum of the range will affect the number of tuples which the range covers. For example, the range {price (15000, 20000)} covers more tuples than that the range {price (0, 5000)} does. In this case, although the sizes of the ranges are the same, the positions of the ranges are different. The more tuples the query range covers, the higher the cost using the MinCS is, because more tuples will return. On the other hand, the cost of using MaxCS will not be affected by the value distribution.

F5: The size of a local database. The larger the size of a local database is, the higher the cost is whatever the strategy is chosen. This is because that more tuples need to be processed and more tuples need to be transmitted and filtered.

F6: Network transmission. The network traffic tends to follow some pattern. As we test the speed of the network at different time in a day, the speed of the network varies over time. Since the intermediate results need to be transmitted to the user, the network traffic would affect the data transmission cost and subsequently the total cost. When the network traffic is bad, using MaxCS will be affected more significantly because the size of intermediate result is larger.

F7: The performance of a local database. Factors of this type include CPU, I/O and memory buffers, etc. of the web database server. These factors affect the query cost obviously but they are difficult to measure.

F8: The size of the intermediate result. The smaller the size of the intermediate result is, the lower the transmission cost would be.

F9: The length of a record. Some returned results contain intermediate information so the tuple size may be large. For the same size of the intermediate result, the larger the tuple size of the result, the lower the filtering cost would be because fewer tuples would need to be filtered.

Some of these factors, which are verified to affect the *Relative Cost* significantly, will be kept as the independent variables X_1, X_2, …, X_k of the regression model.

4.3 Classification of the Global Range Query

Usually, a global range query consists of a set of attributes and their values. For example, {<Make, “Audi”>, <Model, “A6”>, <Price, “50000-60000”>}. Combination with different types of attributes may affect the query cost and the selection of query translation strategy. The attributes besides the range attribute can be classified into three types as follows:

1. Functional attribute: A functional attribute does not affect the retrieval of results other than their display. For example, *Order By* is a functional attribute. So whether the global query contains functional attributes or not, the Relative Cost will not be affected.
2. Categorical attribute: A categorical attribute will limit the result of a query. This is because, usually, categorical attributes are implemented as a selection list and only one value in the list is selected in a typical query. On the other hand, database designers usually create another table to store the categorical values in order to preserve the normalization of the database. If that is the case, the join operation will increase the retrieving cost on the server side of the web database. In summary, if the global range query contains categorical attribute, the Relative Cost may be affected.
3. Value attribute: Such an attribute has many values and only one of them is used to specify an equality condition. A query involving a value attribute will result in very small result set, leading to very low transmission cost and filtering cost.

The types of a query are defined as follows.

Definition 3.5. The *type I query* is a range query that only contains functional attributes in addition to the numeric range attribute.

Definition 3.6. The *type II query* is a range query that only contains categorical attributes in addition to the functional attribute and the numeric range attribute.

Definition 3.7. The *type III query* is a range query that contains value attributes in addition to the numeric range attribute**.**

In order to make the fitness of the regression model better, we construct one cost model for each type of query.

4.4 Pre-processing-based Stepwise Method

For building a multiple regression model, there are two common methods proposed: forward method, backward method [11].

Forward method firstly makes the regression model contain only a constant and then adds the independent variables one by one. However, without considering the correlated relationship between the independent variables, this method may make the independent variables that have been added to the regression model become unimportant later.

Backward method is opposite to forward method. From the beginning, it adds all the independent variables to the regression model and then eliminates the independent variables one by one. However, some independent variables that have been eliminated from the regression model may become relatively important later. So the final regression model may miss some independent variables that are relatively important to the dependent variable.

In this paper, we propose a Pre-processing-based Stepwise Method (PSM) to select significant independent variables into the regression model. PSM is illustrated in Figure 3.

1. PSM firstly analyzes the correlations between each independent variable and the dependent variable by *Pearson correlation coefficient* [11]. The closer the absolute value of *Pearson Correlation Coefficient (PCC)* is to 1, the more highly correlated the two variables are. The correlation can also be observed by the scatter plot. For example, in Figure 2, we can see that RC and Range have a linear relationship.

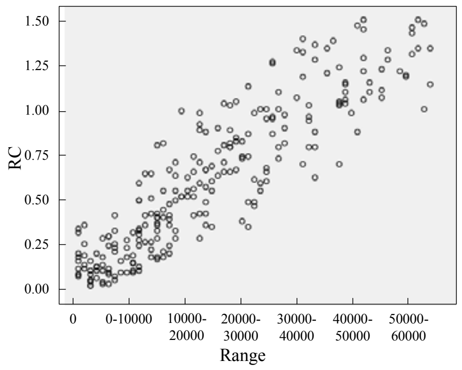

Figure 2. The scatter plot of RC and Range

2. Then it discards those independent variables that have no correlation with the dependent variable and transforms some independent variables that have non-linear correlation with the dependent variable into linear correlation through some functional transformations as much as possible. For example, as we observed from the scatter plot of RC and position, although RC and the position of the query range have non-linear correlation on the whole, they have positive linear correlation between 0 and 30000 and negative linear correlation between 30000 and 60000 on the position. If we

```
Algorithm PSM for MrCoM
Input: the initial set of independent variables (V);
       the set of independent variables (W) in MrCoM;
       the observed data of sample queries;
Output: a Multiple-regression Cost Model (MrCoM)
PSM method:
1.  Begin
2.    Pre-process;
3.    W=Φ
4.    While W is changed Do;
5.      Add;
6.      Eliminate;
7.    End While
8.    Return the current model as the MrCoM
10. End
Procedure Pre-process:
1.  Begin
2.   FOR each variable $X_j$ in V
3.   Analyze the relationship with the responsible variables
4.     If absolute value of PCC is larger than 0.75
5.       Then linear correlation and keep $X_j$
6.     End If
7.     If absolute value of PCC is between 0.25 to 0.75
8.       Then non-linear correlation and try to transform
              $X_j$ into linear correlation
9.       Else discard $X_j$
10.    End If
11.  End For
10. End
Procedure Add:
1.  Begin
2.   For each $X_j$ in V Do
3.     Calculate the significance ($X_j$)
4.   End For
5.   Select Max (significance ($X_k$))   (1<=k<=i)
6.   If (significance ($X_k$) < $a_{in}$
7.     W=W + {$X_k$}
8.     V=V - {$X_k$}
9.   End If
10. End
Procedure Eliminate:
1. Begin
2.  For each variables $X_s$ in W Do
3.   Calculate the significance ($X_s$)
4.   If (significance ($X_s$)) > $a_{out}$
5.       W=W - {$X_s$}
6.       V=V + {$X_s$}
7.   End If
8.  End For
9. End
```

Figure 3. Algorithm PSM

transform the position using the following formula according to the symmetry of the distribution of the scatter plot, RC and the position will have linear correlation.

$$position = \begin{cases} position_0, & position \le 30000 \\ 60000 - position_0, & position > 30000 \end{cases} \quad (8)$$

3. Finally it adds the pre-processed independent variables that have some linear correlation with the dependent variable into the regression model one by one. Note that the significance of X_k is the P-value calculated by using a t-test of a null hypothesis on the correlation coefficient [11]. While a significant independent variable is added to the regression model, the old independent variables will be tested one by one and the independent variable that is changed into an insignificant one will be eliminated from the regression model. On the other hand, any eliminated independent variable that becomes important will be added. So the regression model will not only avoid leaving out any significant independent variables but also avoid containing any insignificant independent variables.

It is important to select the thresholds of significance level a_{in} and a_{out}. Larger a_{in} and a_{out} would lead to more independent variables to be included in the regression model. On the contrary, smaller a_{in} and a_{out} would cause fewer independent variables to be included in the regression model. The values of a_{in} and a_{ou} will influence the precision of prediction of the best query translation strategy. So the appropriate values should be selected for a_{in} and a_{ou}. Our experiments show that $a_{in} = 0.05$ and $a_{out} = 0.10$ are appropriate.

4.5 The Prototype System

The overview of the prototype system is showed in Figure 4. The flow of constructing MrCoM consists of five parts: query sampler, query classifier, factor analyzer, MrCoM builder and MrCoM. The flow of selecting query translation strategy consists of four parts: query classifier, factor analyzer, MrCoM and strategy selector.

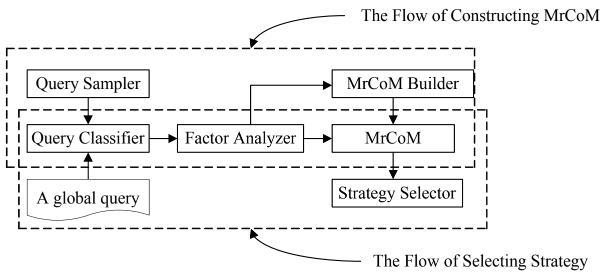

Figure 4. The prototype system

According to the principle of stratified sampling, the query sampler generates a great number of sample queries. The query classifier analyzes the sample queries and classifies them into three types: type I query, type II query and type III query. The factor analyzer records the costs of the two strategies and the values of various factors such as the size of range, the submitting time, etc., which are the sample data for building model, and MrCoM builder constructs the multiple regression model for each class of query. When a new global query is submitted, the query classifier firstly determines which type it belongs to, the factor analyzer records its features, then the RC is computed by MrCoM and the strategy selector determines which strategy is better.

5. EXPERIMENTS

The experiment consists of two parts: (1) build the MrCoM, (2) use MrCoM to choose strategy for query translation.

We manually collect 30 Web sites in cars domain mainly from UIUC data sets [12]. The query interfaces of these sites have numeric range attribute in common. In one experiment, 25 sites are used for building the Multiple-regression Cost Model and the remaining 5 sites are used for verifying the accuracy of the model.

To improve our experimental method, we also utilize 6-fold cross validation to generate the final cost model. The experimental results verify the effective of our method.

5.1 Building the MrCoM

The prototype system accepts a sample query which contains a numeric range attribute (for example, price between 10000 to 40000) and translates it into a local query of a Web database by the two strategies, MinCS and MaxCS, respectively. Then the RC is computed and the corresponding factors are recorded. Given different factors discussed in section 4.2, we adopt the stratified sampling method on the whole and random sampling method in each of the stratification for generating the query sample.

We have discussed the classification of the global query in section 4.3. For each type of query (i.e., type Ⅰ queries, type Ⅱ queries and type Ⅲ queries), we generate about 5000 sample data including the factors and corresponding RC, which have been verified to be sufficient for building the multiple regression model in a previous work [11]. Then we output these sample data into SPSS (Statistical Package for the Social Science) for the statistical analysis and building the regression model.

5.1.1 Building the MrCoM for type I queries

The results of the MrCoM for type I queries are shown in Table 1 and Table 2. From Table 1, we can see that the values of R^2 are 0.739, 0.780, 0.806 and 0.819, respectively, for the four cases. From Table 2, we can see that the standard errors are 0.064, 0.008, 0.015 0.050 and 0.017 for the four cases, respectively. Moreover, in addition to R^2 and the standard errors, the significance of a model can be tested by statistical hypothesis testing such as the t-test. From Table 2, the significant values are 0.012, 0, 0, 0.010 and 0.038 by the t-test. These all indicate that the fitness of MrCoM is good. According to the model coefficients in Table 2, the MrCoM for type I queries is constructed as follows:

$$RC=0.162+0.192X_1+0.101X_2-0.129X_3-0.035X_4 \quad (9)$$

X_1 denotes the range of a global query, X_2 denotes the position of the global query, X_3 denotes the time segment when the query is submitted, and X_4 denotes the granularity of the local range attribute. From the results, X_1 and X_2 are positively correlated with RC while X_3 and X_4 are negatively correlated with RC.

Table 1. Model summary for type I queries

Model	R	R^2	Adjusted R^2	Std. Error of the Estimate
1	.850(a)	.739	.738	.21088
2	.883(b)	.780	.778	.19424
3	.896(c)	.806	.803	.18521
4	.902(d)	.819	.817	.18120

a. Predictors: (Constant), range
b. Predictors: (Constant), range, position
c. Predictors: (Constant), range, position, granularity
d. Predictors: (Constant), range, position, granularity, time

Other factors are excluded from the model, some of which cannot be obtained. (e.g., the performance of a local database) and some of which are insignificant to the model (e.g., the length of the records, because the styles of the tuples in most web sites are similar, usually providing a picture of the entity). Note that we reduced the range, position and granularity by 1,000 times in order to make the coefficients of the model not too small.

Table 2. Model coefficients for type I queries

	Model	Coefficients	Std. Error	t	Sig.
1	(Constant)	.154	.022	7.049	.000
	range	.213	.007	29.363	.000
2	(Constant)	-.030	.032	-0.952	.000
	range	.186	.008	24.493	.000
	position	.112	.015	7.438	.000
3	(Constant)	-.095	.056	1.700	.090
	range	.190	.008	24.759	.000
	position	.102	.018	6.696	.000
	granularity	-.136	.050	-2.719	.007
4	(Constant)	.162	.064	2.518	.012
	range	.192	.008	24.982	.000
	position	.101	.015	6.652	.000
	granularity	-.129	.050	-2.585	.010
	time	-.035	.017	-2.085	.038
a. Dependent Variable: RC					

5.1.2 *Building the MrCoM for type II queries*

Similarly, the results of the MrCoM for type II queries are shown in Table 3 and Table 4. The MrCoM for type II queries is constructed as follows:

$$RC= 0.568+0.221X_1-0.142X_2-0.133X_3 \qquad (10)$$

Table 3. Model summary for type II queries

Model	R	R^2	Adjusted R^2	Std. Error of the Estimate
1	.896(a)	.802	.801	.17299
2	.911(b)	.830	.828	.16067
3	.918(c)	.842	.843	.15503
a. Predictors: (Constant), range b. Predictors: (Constant), range, time c. Predictors: (Constant), range, time, granularity				

Table 4. Model coefficients for type II queries

	Model	Coefficients	Std. Error	t	Sig.
1	(Constant)	. 128	. 019	6.653	.000
	range	.220	.007	31.241	.000
2	(Constant)	.449	.054	8.250	.000
	range	.220	.007	33.516	.000
	time	-.129	.021	-6.252	.000
3	(Constant)	.568	.059	9.626	.000
	range	.221	.006	34.907	.000
	time	-.142	.020	-7.066	.000
	granularity	-.133	.030	-4.402	.014
a. Dependent Variable: RC					

where X_1 denotes the range of a global query, X_2 denotes the time segment the query is submitted, and X_3 denotes the position of the local range attribute. Compared with the MrCoM of type I queries, the independent variable denoting the position of the global query is insignificant to the model. Intuitively, for different values of the categorical attribute, the distributions of the tuples on the numeric range attribute are greatly different.

5.1.3 *Building the MrCoM for type III queries*

For queries of type III, because they contain the *value attributes*, the results are usually small in size. In general, the entire results can usually be displayed in one page even when MaxCS is used. For example, the results of the query {<Keywords, "CHEVROLET SEDAN">, <Price, all>} for Dealsonwheels.com contain only 5 records. In contrast, when using MinCS, the results may be displayed in several pages, one for each of the queries submitted to the local Web database. Because the retrieving cost of MinCS is usually higher while the transmission cost and filtering cost are not obviously lower than the costs of MaxCS for type III queries, the overall cost of MinCS is usually higher than that of MaxCS for type III queries. This is confirmed by our experimental results, which show that most values of RC for type III queries are larger than 1.

5.1.4 *6-Fold Cross Validation*

The above experimental method is the simplest kind of cross validation. The data set is separated into two sets, called the training set (25 Web sites) and the testing set (5 Web sites). However, the evaluation may be significantly different depending on how the division is made.

We use 6-fold cross validation to improve our experimental method. We divide the 30 Web sites into 6 subsets, each of which contains 5 Web sites. Each time, one of the 6 subsets is used as the test set for verifying the accuracy of the cost model and the other 5 subsets are put together to form a training set for generating the cost model. 6-fold cross validation method is repeated 6 times and the average model coefficients for each type of queries are computed. The final model is shown in Table 5. In the MrCoM for type I queries, X_1, X_2, X_3 and X_4 denote the range, the position, the time and the granularity respectively. In the MrCoM for type II queries, X_1, X_2 and X_3 denote the range, the time and the granularity respectively. The advantage of this improved method is that it matters less how the data gets divided.

Table 5. MrCoM using 6-fold cross validation

MrCoM for type I queries
$RC=0.168+0.190X_1+0.106X_2-0.141X_3-0.038X_4$
MrCoM for type II queries
$RC= 0.551+0.223X_1-0.137X_2-0.133X_3$

5.2 Performance Metric for MrCoM

For a given global query, the prototype system computes the RC by MrCoM and selects the better translation strategy. On the other hand, we summit the query to the Web database by the two strategies and choose the one that actually costs less. If the strategies chosen by the two methods are consistent, it indicates that the model is accurate.

$$accuracy=\frac{the \quad number \quad of \quad consistent \quad strategy \quad selections}{the \quad number \quad of \quad queries} \qquad (11)$$

In this paper, the accuracy metric is used to measure the effectiveness of MrCoM.

5.3 Experimental Results

For the 5 testing Web databases which are random selected from 30 Web databases, we also adopt the stratified sampling method on the whole and random sampling method in each of the stratification for generating the query sample. We design close to 300 global range queries of each type for a Web database (i.e., totally close to 4500 global queries) as illustrated in Table 6.

Table 6. An instance of random design of the global queries

Range (<)	Position (in)	Time (in)	Number			Total Number
10,000	0-10000	0:00-3:00	1	8	48	288
		3:00-6:00	1			
		6:00-9:00	1			
		9:00-12:00	1			
		12:00-15:00	1			
		15:00-18:00	1			
		18:00-21:00	1			
		21:00-24:00	1			
	10000-20000	…	…	8		
	20000-30000	…	…	8		
	30000-40000	…	…	8		
	40000-50000	…	…	8		
	50000-60000	…	…	8		
20000	…	…	…	…	48	
30000	…	…	…	…	48	
40000	…	…	…	…	48	
50000	…	…	…	…	48	
60000	…	…	…	…	48	

Figure 4 shows the result of experiments. As we can see, the accuracy of our model is generally high. In details, the average accuracy of MrCoM for type I queries, type II queries and type III queries are 92.9%, 91.5% and 97.6% respectively.

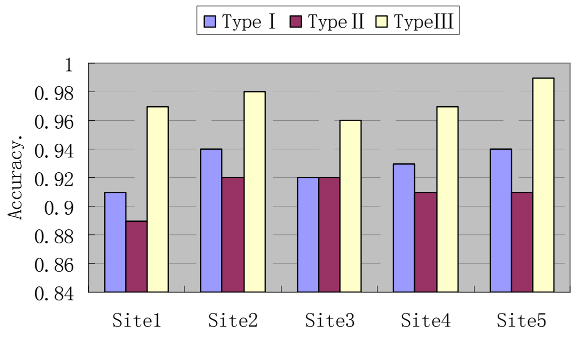

Figure 4. Strategy selection accuracy on 5 web databases

Given that the MrCoM in our approach is constructed for one domain, not just for individual site, we believe it is practically feasible in real life Web databases.

6. CONCLUSIONS

A major challenge for query translation in Deep Web Data Integration System is the lack of cost models at the global level. To tackle this challenge, this paper proposed the Multiple-regression Cost Model (MrCoM) based on statistical analysis for the global range queries that contains numeric range attributes. We also proposed the Pre-processing-based Stepwise Algorithm (PSA) for selecting significant independent variables into the MrCoM. Experimental results demonstrated that the fitness of the obtained MrCoM using our approach is good and the accuracy of the query strategy selection is high.

7. ACKNOWLEDGE

This research was partially supported by NSFC under grant No. 60573091, National High-Tech Research and Development Plan under grant No. 2007AA01Z155, Program for New Century Excellent Talents in University (NCET), and US NSF grants IIS-0414981 and CNS-0454298.

8. REFERENCES

[1] M. Bergman. The Deep Web: Surfing Hidden Values. White Paper of CompletePlanet. 2001. (Available at http://brightplanet.com/pdf/deepwebwhitepaper.pdf)

[2] Q. Zhu, and Larson, P. A Query Sampling Method of Estimating Local Cost Parameters in a Multidatabase System. ICDE Conference, Houston, USA, 1994, pp.14-18.

[3] S. Raghavan, and H. Garcia-Molina. Crawling the Hidden Web. VLDB Conference 2001, Roma, pp.129-138.

[4] P. Wu, J. R. Wen, H. Liu, and W. Y. Ma. Query Selection Techniques for Efficient Crawling of Structured Web Sources. ICDE Conference 2006. pp. 47.

[5] H. He, W. Meng, C.Yu, and Z. Wu. Automatic Integration of Web Search Interfaces with WISE-Integrator. Journal of Very Large DataBase. 13(3), September 2004, pp.256-273.

[6] K. C. C. Chang, B. He and Z. Zhang. Toward Large Scale Integration: Building a MetaQuerier over Databases on the Web. CIDR Conference 2005, Asilomar, USA, pp.44-55.

[7] J. Wang, and F. H. Lochovsky. Data extraction and label assignment for web databases. WWW Conference 2004, New York, USA, pp.187-196.

[8] A. Arasu, H. Garcia-Molina. Extracting Structured Data from Web Pages. ICDE Conference 2003, Bangalore, India. pp. 698-698.

[9] Q. Zhu, and P. Larson. Solving Local Cost Estimation Problem for Global Query Optimization in Multidatabase Systems. Distributed and Parallel Databases, 6(4), 1998, pp. 373-421.

[10] Z. Zhang, B. He, and K.C.C. Chang. Light-weight Domain-based Form Assistant: Querying Web Databases On the Fly. VLDB Conference 2005, Trondheim, Norway, pp. 97-108.

[11] R. C. Pfaffenberger, and Patterson, J. H. Statistical Methods for Business and Economics. Richard D.Irwin. 1987.

[12] http://metequerier.cs.uiuc.edu/repository/