I.J. Information Technology and Computer Science, **2010, 1, 32-39**
Published Online November 2010 in MECS (http://www.mecs-press.org/)

A Genetic Programming Framework for Topic Discovery from Online Digital Library

Yinxing Li
Province Research Institution of Regional Economy
Beihua University, Jilin, PR. China
Email: fnyjjgl@126.com

Ning Li
School of Economics and Management
China University of Petroleum, Dongying, China
Email: liningmath@163.com

***Abstract*—Various topic extraction techniques for digital libraries have been proposed over the past decade. Generally the topic extraction system requires a large number of features and complicated lexical analysis. While these features and analysis are effective to represent the statistical characteristics of the document, they didn't capture the high level semantics. In this paper, we present a new approach for topic extraction. Our approach combines user's click stream data with traditional lexical analysis. From our point of view, the user's click stream directly reflects human understanding of the high-level semantics in the document. Furthermore, a simple, yet effective, piece-wise linear model for topic evolution is proposed. We apply genetic algorithm to estimate the model and extract topics. Experiments on the set of US congress digital library documents demonstrate that our approach achieves better accuracy for the topic extraction than traditional methods.**

***Index Terms*—Genetic Algorithms; Non-linear Matrix Factorization; Web-click Data; Convex Optimization; Interior Point Method.**

I. Introduction

Today digital libraries and web information system become a rich and open resource for researchers. Those systems cover various research topics in many different subjects. More and more researchers have found it is convenient to find interesting research topics, seminal research papers and latest progress in research frontiers using those systems. The increasing number of people are contributing to those systems, either deliberately or incidentally. This has created a huge set of data that could potentially be used to improve the services of digital library and web information system in general. On the other hand, the phenomenon of the wide adoption of on-line digital library and web information system as a primary means for knowledge sharing has led to a great booming of research in this field over the past 50 years. A lot of innovative tools and clever algorithms have been developed to facilitate the effective and efficient usage of these systems, for instances document indexing [3], information retrieval [4], document classification [10] etc.

On the other hand, with the prevailing publishing and subscribing services provided by the internet, digital libraries and web information systems cover various topics in many different research subjects [2]. Indeed, they have become such a rich and open resource available to research communities. Through them, researchers can easily access a great amount and a wide variety of information. With those information, it is convenient to find interesting research topics, seminal research papers and latest progress in research frontiers. The phenomenon of the wide adoption of on-line digital library and web information system as a primary means for knowledge sharing led to a great booming of research in this field over the past 50 years. A lot of innovative tools and clever algorithms have been developed to facilitate the effective and efficient usage of these systems, for instances document indexing [3], information retrieval [4], document classification [10] etc.

Recently, Web 2.0, the new generation of web, has become increasingly popular. The important characteristic for Web 2.0 is the interactive usage of the web content and the wealth of useful meta-data. Unlike traditional web, in which most users passively receive information; in Web 2.0, the users are playing a much more interactive role. When browsing the web, the users perform many different actions, such as bookmarking website, tagging documents with key words, making recommendations to friends, and sharing documents among collogues etc. The new concept and technology of

"Web 2.0" has introduced other innovative usages and applications for the web. Examples include semantic web, personal blog, social networks etc. [9]. Digital library, as a special kind of web information system, are also influenced by the new Web 2.0 technology. Researchers have recently started to take advantage of Web 2.0 and incorporate the meta data to improve the services of digital library systems. For example, the authors [10] use the history tagging information to make recommendation and suggestion on future articles for the users.

"Topic discovery" is one of the most important applications of digital libraries [3] [4][10]. It is useful to understand the landscape of research area, discover

potential research directions, and predict the research breakthrough in future. However, the problem of "topic discovery" is very challenging because of (i) the huge gap between low level feature and high level semantics and (ii) the continuously changing of the research topics over time. Traditional approaches focus on analyzing and detecting topics using document content features and document structure analysis [1]. While these features and analysis are capable to represent the statistical characteristics of the documents, they didn't capture the high level semantic content of the document. And they often require a large amount of training data in order to be effective. This further limits their applicability for tracking new research topics, where the training data is often sparse. Fortunately, the interactive usage of web 2.0 generates lots of user data, which partially helps to solve the sparsity of training data problem. One of the simple yet effective data in online systems are the Click Stream. Those data are recorded from the user's browsing session and stored in form of web-logs. They can be easily accessed through meta-data attributes associated with the document. They are highly correlated to end user's perception and closely follow the research trend development. These characteristics make them ideal candidate to improve the performance of "topic discovery" for online digital library. In this paper, we present a new approach to incorporate the Click Stream data for "topic discovery".

TABLE I. A SNAPSHOT OF USER'S CLICK STREAM DATA.

Time	UserID	Query	DocLink	Summary
5/11/09 9:03	198	Nuclear Weapon	3758.html	…
5/11/09 9:07	198	Nuclear Weapon	4532.html	…
5/11/09 9:18	198	Nuclear Weapon	2741.html	…
5/18/09 10:03	662	Health Insurance	6856.html	…
5/18/09 10:11	662	Health Insurance	7423.html	…

A. *Background and Motivation*

A snapshot of the click stream data is shown in Table I. The click stream contains a list of five-item tuples. Each tuple records 5 different pieces of information about the user's query: (1) the time stamp for query, (2) the anonymous user's identification number, (3) the query phrase issued by the user, (4) the document link that the user clicked and (5) the summary of the clicked document (usually the abstract of the document).

Before going into details on how to process the click stream data, we explain the motivation for using the aggregated click stream data as a promising source for the task of topic discovery. Firstly, user's clicks represent direct measurement for topic relevance from human's point of view. The most recent clicked documents often represent the latest research frontier and progress in the field. Secondly, the click stream data are well-formated and logged automatically by the system. This leads to large volume of data in a short time, which is very important for comprehensive knowledge discovery and timely tracking of the new research trend. Thirdly, unlike the traditional document content data, which requires expensive preprocessing, these log data could be easily and efficiently processed.

However, topic discovery from the click stream is still a non-trivial problem despite the simplicity of the data. This is because: (1) the information is incomplete. There are only a very limited amount of semantic contents available (i.e. abstract etc.). (2) There are a large amount of noise in the click stream data, which doesn't represent any relevant topics.

Our research on using the click stream data for topic discovery in on-line digital library is motivated by the event detection in the web [11]. In [11], the authors presented a clustering based approach for web event detection. In their works, they use a bottom-up agglomerative clustering algorithm that groups the click stream data based on their semantics. The clustering outputs are detected as web events. Their approach may be suitable for web system, but not adequate for digital library system. We notice that their approach ignores the actual document content completely and rely solely on pair-wire distance of click stream data for clustering. However, the content of the articles in digital library is well organized. The abstract provides a strong clue for grouping topics though it is short and fast to process. Therefore, we combine both abstract and click stream in our approach for topic discovery. Furthermore, rather than using a non-parametric cluster approach [11], we present a model based approach for topic discovery. An explicit piece-wise linear function is proposed for modeling the evolving of research topics. Every topic is modeled as a linear segment along the topic evolution curve. The model is simple, but effective. The piece-wise linear function continuously follows and monitors recent research progress. To overcome the limitation of the linear modeling, a nonlinear optimization algorithm, genetic algorithm, is applied to estimate and extract the linear segments from the topic curves.

The following of the paper is organized as follows. In Section II, we present the feature processing on click stream data. We describe the piece-wise linear model for the topic evolution and the genetic algorithm to estimate and extract topics in Section III. Experimental results based on the library of US congress document database are presented in Section IV. We conclude the paper in Section V.

II. FEATURE EXTRACTION AND REPRESENTATION

In this section, we present feature processing for online click stream data.

A. Preprocessing

Table I shows the raw click stream accumulated from the on-line digital libraries. In the table, each row records a single round of query and retrieval, (i.e. query-

document pair generated by a single click). Each query record corresponds to a single round of query and retrieval. However, instead of considering individual query records, we preprocess the data by aggregation. In preprocessing, individual query records are grouped into larger query groups. Each query group consists of multiple query records issued by the same user within a short time interval. This preprocessing is based on the observation that a user is often interested in one topic within a short time period. Therefore, multiple queries submitted consecutively by the same user within a short time is likely to be on the same topics. The query group provides a set of highly related query-document pairs and exhibits better semantic coherence. The query group is defined as follows:

$$G = \left[t, \{q_1, q_2, ..., q_m\}, \{\alpha_1, \alpha_2, ..., \alpha_m\} \right]$$

where t records the first query submission time, $\{q_1, ..., q_m\}$ refers to the set of query key words, $\{a_1, a_2, ..., a_m\}$ represents the abstracts of the relevant documents. Referring to Table I, the first three rows are considered to be the query group G_1 of "Nuclear Weapon" as they are submitted by user 198 within a short time interval (1m). The last two rows are considered to belong to the query group G_2 of "Health Insurance" submitted by user 662.

B. Feature Extraction

In our approach, we find an embedding feature vector for each query group. This process involves two steps: (1) compute the pair-wise similarity matrix between any two query groups and (2) find embedding feature vector for each query group using Non-negative Matrix Factorization.

1) Similarity matrix: The pair-wise similarity matrix ($\boldsymbol{S_{i,j}}$) are computed from key words ($\boldsymbol{Q_i = \{q_{i,1}, ..., q_{i,m}\}}$) and abstracts ($\boldsymbol{A_i = \{a_{i,1}, ..., a_{i,m}\}}$) for query groups $\{\boldsymbol{G_1, G_2, ..., G_n}\}$. Both $\boldsymbol{Q_i}$ and $\boldsymbol{A_i}$ are modeled as bag-of-words. The similarity between two groups $\boldsymbol{G_i}$ and $\boldsymbol{G_j}$ is defined as the weighted sum of the Jaccobi coefficients for query key words and documents (equation (1)).

$$S_{i,j} = (1-\alpha) \cdot \frac{\left| Q_i \text{ I } Q_j \right|}{\max\left(\left| Q_i \right|, \left| Q_j \right| \right)} + \alpha \cdot \frac{\left| D_i \text{ I } D_j \right|}{\max\left(\left| D_i \right|, \left| D_j \right| \right)} \quad (1)$$

where ∩ denotes the intersection of two sets, | · | refers to the size of the set, and α is the weighting coefficient. In our implementation, we choose α to 0.5, which gives equal weights on the click stream and abstract data.

2) Feature projection with NMF: From the similarity matrix $\boldsymbol{S}$, we compute a linear projection that embeds the query groups into a lower dimension space. Non-negative Matrix Factorization (NMF) [7], [8] are applied for this task.

Similar to other linear dimension reduction algorithms (i.e. PCA, ICA etc.), NMF finds a set of linear spaces to embed the original data. However, NMF enforces the nonnegative constraint, where each entry in factorization matrices must be greater than or equal to zero. This constraint is applicable to models where the negative value or subtraction doesn't make sense. For our problem, each entry in the feature vectors, modelled as bag-of-words, denotes the word count and negative value and subtraction is not meaningful. Therefore, NMF is more suitable for our problem than other similar approaches (i.e. PCA).

In the following, details for feature embedding using NMF are presented. The NMF problem can be formulated as follow:

Given matrix M, find two non-negative matrices U and V, such that

$$M_{ij} \approx (U,V)_{ij} = \sum_{k=1}^{r} U_{i,k} V_{k,j} \quad (2)$$

where the r columns in $\boldsymbol{U}$ represent the NMF linear spaces, and the columns in $\boldsymbol{V}$ are the combination coefficients, which is the result after dimension reduction. $\boldsymbol{M}$ is $\boldsymbol{m \times n}$, $\boldsymbol{U}$ is $\boldsymbol{m \times r}$ and $\boldsymbol{V}$ is $\boldsymbol{r \times n}$. Here m is the dimension of the original feature, n is the number of samples, r is the number of reduced dimensions.

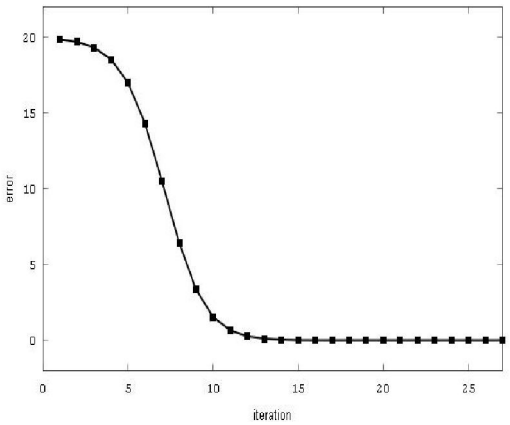


Figure 1. Convex Solver for the NMF problem of matrix size 30 × 100.

In equation (2), direct NMF requires to know the actual m-dimensional feature vector for each sample and stack them into matrix **M**. However, we only know pair-wise similarity matrix **S** (equation (1)) for our problem, not the direct feature matrix **M**. This problem can be solved by using the kernel trick. Considering $\mathbf{M_j}$, we introduce the $\Phi(\mathbf{M_j})$, which is a nonlinear mapping from the original input space **M** to a higher, possibly infinite, dimension feature space **F**:

$$\phi : x \in M \rightarrow \phi(x) \in F \quad (3)$$

The kernel mapping $\mathbf{M_\Phi}$ and $\mathbf{U_\Phi}$ for the stacked matrix **M** with n samples and the NMF base matrix **U** with r basis are denoted as

$$M_\phi = \left[\phi(M_1), \phi(M_2), ..., \phi(M_n) \right] \quad (4)$$

$$U_\phi = \left[\phi(U_1), \phi(U_2), ..., \phi(U_n) \right] \quad (5)$$

Then NMF equation are rewritten as

$$M_\phi = U_\phi \cdot H \quad (6)$$

Same as in equation (2), $\mathbf{U_\Phi}$ and $\mathbf{H}$ are bases and combining coefficients. Each column of $\mathbf{H}$ is the reduced feature vector. However, it is impossible to directly compute $\mathbf{H}$ from equation (6) because we don't know $\mathbf{M_\Phi}$ yet. Multiplying the equation (6) by $\mathbf{M_\Phi}^T$, we obtain

$$\left[M_\phi{}^T M_\phi\right] = \left[M_\phi{}^T U_\phi\right] \cdot H \tag{7}$$

$[\mathbf{M_\Phi}^T \mathbf{M_\Phi}]$ calculates the inner product between two samples in the projected kernel space, $\mathbf{S_k}$. $[\mathbf{M_\Phi}^T\mathbf{U_\Phi}]$ denotes the inner product between feature samples and bases in the kernel space. In our implementations, we choose to use the Radial Basis Function kernel function (equation (8)).

$$S_k(i,j) = \left[M_\phi{}^T M_\phi\right]_{i,j} = \exp(-\gamma S(i,j)) \tag{8}$$

To simplify the notation, let $\mathbf{W}$ denotes $\mathbf{M_\Phi}^T\mathbf{U_\Phi}$. Then, kernel NMF becomes an optimization problem defined (see equation (9)). $\|\cdot\|_n$ denotes the matrix n-norm operations.

$$\begin{aligned} &\min_H \|S_k - WH\|_n \\ &subj.to. \quad W_{i,j} \geq 0, W \text{ is } rank(r) \text{ and } m \times r \\ &\qquad\quad H_{i,j} \geq 0, H \text{ is } rank(r) \text{ and } r \times n \end{aligned} \tag{9}$$

In our implementation, we choose n equal to 2. The objective function becomes a l_2 − norm, which is convex and can be easily solved using convex programming solver (i.e. interior point method [5] etc.).

Fig. 1 shows the performance of the convex solver of NMF for an example problem with matrix size 30×100. As figure 1 shows, the algorithm converges quickly. In practice, we found that it usually took 20-40 iterations to achieve convergence.

III. TOPIC EVOLUTION MODELING AND EXTRACTION

In previous section, we have talked about how to extract features from users' click stream data and use NMF to project the features into lower dimension space. After these steps, In this section, we present our model for topic evolution.

A. Piecewise linear model

In our approach, the topic evolution is modeled as piecewise linear functions. Every topic is modeled as a linear function on the evolving curve. Topic changes happen at the joint of two distinct linear functions. Fig. 2 illustrates the idea of using this model for online topic evolution. In Fig. 2, x-axis represents time and y-axis represents the features. Three distinct topics exist in this time period. The red/green/blue dots correspond to individual feature embedding for query group from three topics. The dotted lines in Fig. 2 represent the piecewise linear model for 3 topics. The topic changes take place at the joint between different linear segments. Equation (10) shows the topic evolution model.

$$y_i = w^T \cdot x_i + b, \quad for\ x_i \in [t_i, t_{i+1}) \tag{10}$$

where x_i denotes the query feature vector, t denotes the time, w and b denotes the weight matrix and the offset respectively.

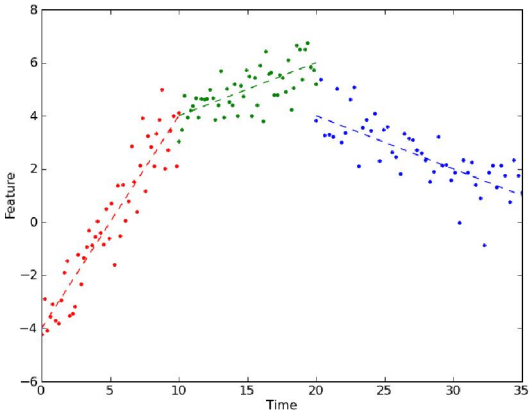


Figure 2. Illustration of piece-wise linear model for topic extraction.

Given the model defined in equation (10), topic extraction becomes a problem of estimating distinct linear functions from the topic curve. The estimation of the piecewise linear function in equation (10) involves two steps: (i) find a partition of the linear curve and (ii) estimate a linear function for each partition. This problem is nonlinear optimization problem and we use a genetic algorithm to solve it.

B. Genetic Algorithm

Genetic algorithm is a high-level algorithm, which is capable of solving a large set of optimization problems. Algorithm 1 shows the structure of genetic algorithm for topic extraction problem.

Algorithm 1 Genetic Algorithm for Topic Extraction

Parameters: Crossover probability: $p_c \in (0,1)$, Mutation probability: $p_m \in (0,1)$.

1: **Initialization:** Generate the initial population P_0 of n individuals with binary encoding chromosome.
2: **Fitness:** Evaluate the fitness of each individual of the population by equation (11).
3: **while (not terminated) do**
4: **Selection:** Select a subset of m pairs of individual chromosomes from the population P_t with the selection operator.
5: **Crossover:** With probability p_c, cross each of the m chosen pair with the crossover method.
6: **Mutation:** With probability p_m, mutate each of the offspring in the result from the crossover with the mutation method.
7: **Fitness:** Evaluate the fitness of each offspring by equation (11).
8: **Replacement:** Create a new generation from individuals in the old generation and off-springs using the replacement strategy defined below.
9: **end while**
10: **Return:** Best found individual chromosome.

Chromosome representation. Each individual chromosome defines a potential partition for the topic functions. We use a binary string based representation, where value "1" represents the change of the topics.

There are two choices for chromosome representation: (i) a list of integers, where each entry defines the topic identity for the query group and (ii) a binary string, where value "1" represents the change of the topics. Example chromosomes of both representations for a 3-topic query sequence are shown below:

Integer	1	1	1	2	2	2	2	3	3	3
Binary	1	0	0	1	0	0	0	1	0	0

Compared with the list-of-integer chromosome representation, the binary string representation is compact and efficient to process. Later on, we will see that it is also more natural and meaningful to define cross-over and mutation operations on binary string representation than on the list-of-integer representation. Therefore, we choose to the binary string based chromosome representation in our implementation,

Fitness function. For a given chromosome, the fitness function is defined as equation (11), where the first term is the sum of the fitting error from all the linear segments, and the second term represents the penalty from incorrect estimation for the number of linear segments.

Note that without this penalty term, the GA algorithm tends to generate too many small topic segments. In equation (11), the constant n_0 represents the prior knowledge about the number of desired topics. It is set by the user. The constant α controls the flexibility for the number of the topics. Larger α means less flexible choice on the number of topics. In practice, we choose α equal to 1% of the total number samples.

$$fitness(g) = \exp\left(-\frac{\sum_{i=1}^{n} \left\| y_i - w^T \cdot x_i - b \right\|_2^2}{\sigma^2} \right) - \alpha \cdot \exp(-|n - n_0|) \quad (11)$$

Selection operators. Selection operators are used to select individual chromosomes to which the crossover operators will be applied. In literature, several selection operators have been proposed, i.e. "Select Random", "Select Best" and "Tournament Selection". In Section IV, we present experimental results to compare the performance of different selector operators.

Crossover operators. Crossover is an important operators in GA algorithm, which select individuals from the parental generation and interchange their genes to generate new individuals (descendants). The aim here is to obtain descendants of better quality that will be propagated to the future generation and enable the search to explore new regions of solution space not explored yet. Many types of crossover operators have been explored in the evolutionary computing literature, which depends on the chromosome representation. For our problem, the crossover operator are defined for the binary string representation. Two crossover operators are considered: (i) uniform crossover (equation 12) and (ii) one-point crossover (equation 13).

Uniform crossover generates a crossover mask and use it to generate one descendant. The crossover mask are defined as $m[i] \in 0/1$, $i \in [1, n]$, where n is the length of chromosome. Once the crossover mask $m[i]$ is generated, the descendent chromosome is computed from two parent p^t_1 and p^t_2 as follows:

$$\forall i = 1, 2, ..., n \;\; p^{t+1}[i] = \begin{cases} p_1^t[i], \; if \;\; m[i] = 1; \\ p_2^t[i], \; if \;\; m[i] = 0; \end{cases} \quad (12)$$

One-point crossover operator randomly chooses a position between 1 and chromosome length, n. The result point served as a "cutting point" which split each parent into two segments. Then, the two first parts of the parents are interchanged to yield two new descendants (equation (13)).

$$\forall i = 1, 2, ..., n \;\; p_1^{t+1}[i] = \begin{cases} p_1^t[i], \; if \;\; i < k; \\ p_2^t[i], \; if \;\; i \geq k; \end{cases} \qquad p_2^{t+1}[i] = \begin{cases} p_2^t[i], \; if \;\; i < k; \\ p_1^t[i], \; if \;\; i \geq k; \end{cases} \quad (13)$$

In our implementation, the crossover operator is applied with probability p_c = 40%. In Section IV, experimental results that compare the performance for different cross operators are presented.

Mutation operators. Several mutation operators based on the binary string chromosome representation are defined, which take into account the specific need of topic extraction. We defined the following operators: ***Move***, ***Merge*** and ***Split***.

Move: This operator moves a partition point forward or backward so that one topic grows and the adjacent topic shrinks. The effect of this operator on the chromosome is to exchange two alleles with one and zero respectively. Note that this operator doesn't change the total number of topics. It is able to explore all possible partitions for a particular number of topics.

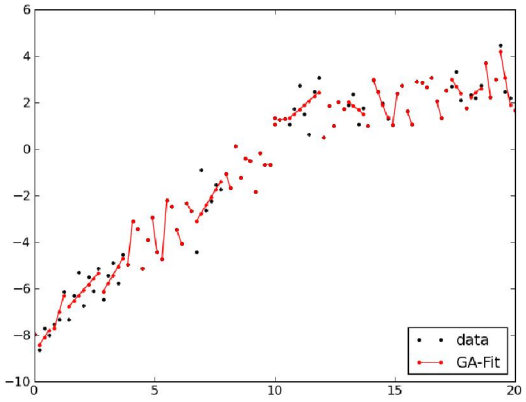
Merge: This operator merges two adjacent topics. The effect of this operator on the chromosome is to change one allele with value one to value zero. This operator will decrease the number of topics by one.

Split: This operator splits one topic into two. The effect of this operator on the chromosome is to change one allele from value zero to value one. This operator will increase the number of topics by one.

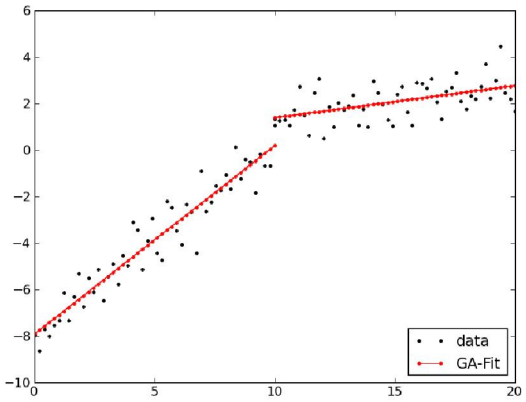
With the three operators, GA algorithm is able to explore all possible partition of the topics. The mutation is applied with a probability p_m = 20%.

Replacement operators. Replacement operator is used to determine how to replace the portion of the parent generation with the individuals from the descendent generation. It defines the survival of the individuals. Let **P** denotes the parent generation and **P'** denote the descendent generation of individuals. μ and λ are the sizes of **P** and **P'**, respectively. The "Replace only if better" strategy is used in our GA algorithm, viz. the new

generations is formed by choosing μ best individual from μ + λ individuals in both the parent generation(**P**) and the descendent generation(**P**'). In our implementation, P_r =20%



(a)



(b)

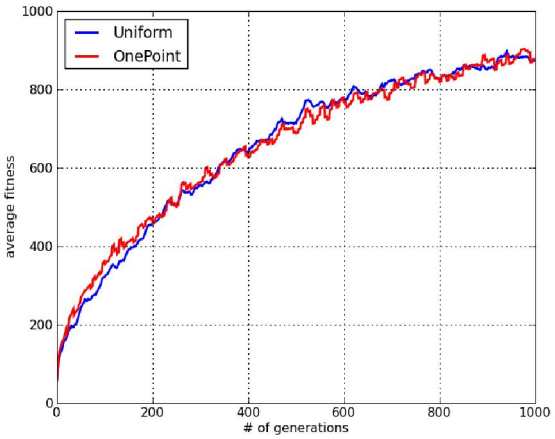
Figure 3. Genetic algorithm for fitting piece-wise linear model. (a) initialize and (b) fitting result after 900 generations.

Fig. 3 demonstrates Algorithm 1 on a set of synthetic data, which consists of two topics. Fig. 3(a) shows an example gene at the beginning of the algorithm. Fig. 3(b) shows the result after 900 generations. It is evident that GA algorithm correctly identifies two topics and fits each topic with a linear function.

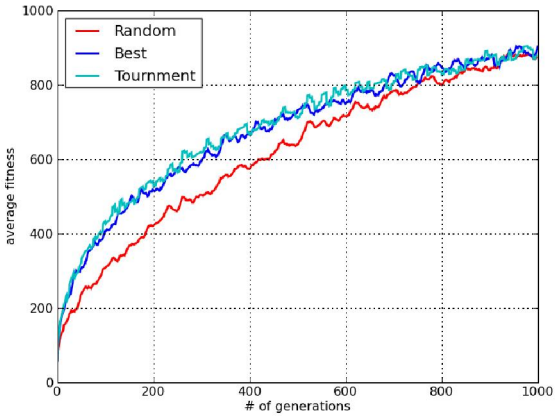
IV. EXPERIMENTAL RESULTS

In this section, the experiments on applying GA algorithm for topic extraction from digital library data are.

Economic	Education	Military	Energy
Federal	Education	Government	Energy
Labor	School	Military	Power
Insurance	Aid	Foreign	Water
Aid	Children	Tax	Nuclear
Tax	Drug	Congress	Gas
Business	Students	Aid	Petrol
Employee	Elementary	Law	Research
Care	prevention	Policy	pollution
		Terrorist	



(a)



(b)

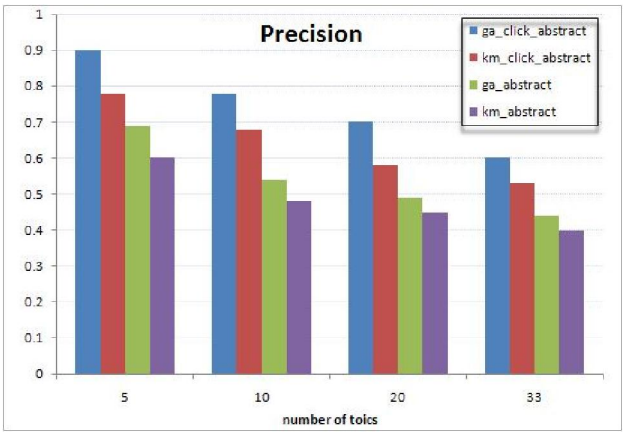
Figure 4. Comparison of performance for different GA operators: (a) selection operators and (b) crossover operators.

We have implemented GA based topic extraction algorithm using the Python Genetic Evolution Algorithm Framework -pyevolve [6]. The fitness function, select operators, crossover operators, mutation operators and replacement operators are defined as plug-in callback functions. The benchmark data are obtained from the US congress document library database (1989-2006) and the query sessions were recorded when the users browsed the library. In total, there are 33 different topics in the database. The user's click stream are grouped into 3432 query sessions and preprocessed into feature vectors as described in Section II. Table II lists the total 33 topics. Those topics fall into four categories: (1) Economic, (2) Education, (3) Military and (4) Energy.

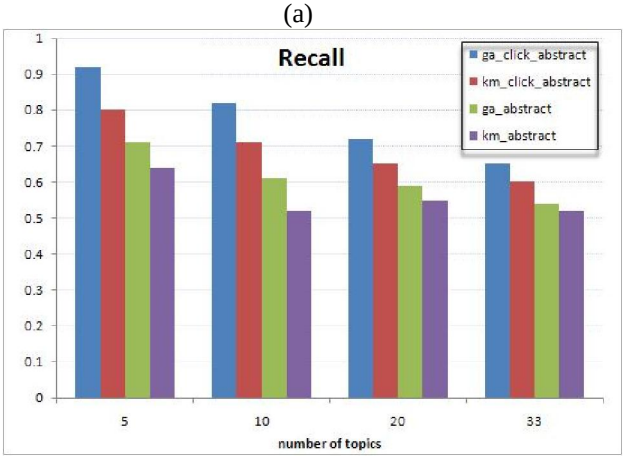
The performance for different choices of GA operators on the benchmark data are presented in Fig. 4. Fig. 4(a) and (b) show the results for three different GA selector operators -Random selection, Best selection, and Tournament selection (N=10), and two different crossover operators - uniform and one crossover, respectively. In both Fig. 4(a) and (b), x-axis is the number of generations and y-axis is the average fitness function over the populations. In Fig. 4(a), the Tournament selector and best selector performs better

than the random selector. There is not significant difference between uniform crossover operation and one-point crossover operators in Fig. 4(b).

Fig. 5 (a) and (b) compares the *precision* and *recall* rates for GA algorithm against the k-means algorithm with abstract-only data (equivalent to $\alpha = 1$ in equation (1)) and the abstract combined with click-stream data. In Fig. 5(a) and (b), the x-axis denotes the number of topics and y-axis denotes the precision and recall rates. It can be seen that for all the dataset with different number of topics (5, 10, 20, 33), GA topic extraction algorithm outperforms the simple k-means algorithm in both precision and recall. The combination of click-stream with document abstract improves the result of document topic discovery. The main reasons for the better performance are (i) the piece-wise linear curve model of the topics evolution and (ii) GA based topic estimation and extraction algorithm.



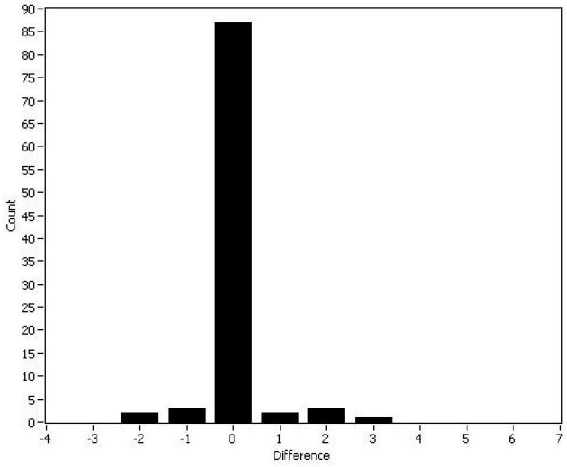
(a)



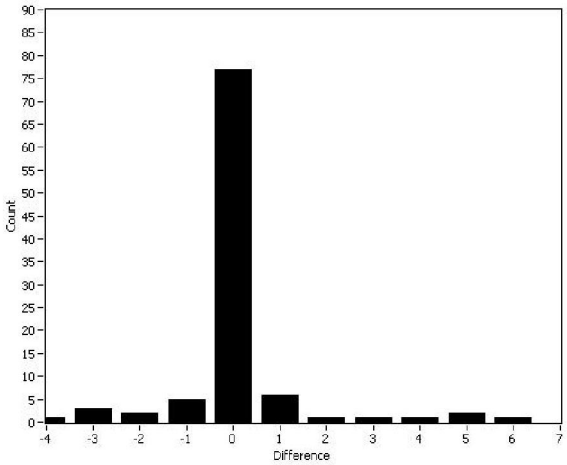
(b)

Figure 5. Comparison of (a) precision and (b) recall rates for GA based vs. k-means algorithm for topic extraction.

Fig. 6 (a) and (b) plot the distribution of the difference between the estimated number of topics and true number of topics for (a) 10-topics and experiment (b) 20-topics in order to study how well GA algorithm performs to extract the correct number of topics. In both settings of experiments (a) and (b), the dataset are obtained by randomly choosing 10 and 20 topics and their feature vectors from Table II. We set initial parameter n in Algorithm 1 to be in range [8, 12] for experiment (a) and [15, 25] for experiment (b). In both experiments, we run the GA algorithm 1000 times with parameter n chosen randomly from the corresponding ranges and compare the estimated number of topics with the true number of topics. From Fig. 6 (a) and (b), it can be found that GA algorithm is capable of find the correct number of topics for about 90% of the cases. This indicates that GA algorithm is robust against the choice of number of topics.



(a)



(b)

Figure 6. Distribution plots for the difference between estimated topic number and true topic number for (a) 10-topics and (b) 20-topics.

V. CONCLUSIONS

User's click stream have recently identified as a potential source for topic extraction. In this paper, we present a GA based approach to extract document topics for online digital library. Our approach combines user's click stream and document abstracts. There are three main contributions for this paper. First, a novel approach to combine click-stream with document abstract to improve the topic discovery process is developed. Secondly, a piece-wise linear function is proposed to model the topic evolution. Thirdly, a Genetic Algorithm is developed for topic extraction. The proposed approach automatically determines the number of topics, and groups query instances and library documents into different (a) (b) topic categories. Experimental results on

US congress documents library demonstrates the effectiveness of the proposed approaches.

REFERENCES

[1] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In SIGIR, 1998.

[2] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information retrieval. Addison-Wesley, 1999.

[3] C. Barry and L. Schamber. Users criteria for relevance evaluation: A cross-situational comparison. Information Processing and Management, 34(2-3):219–236, 1998.

[4] N. J. Belkin. Intelligent information retrieval: Whose intelligence? In Proceedings of the Fifth International Symposium for Information Science, pages 25–31, 1996.

[5] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

[6] http://pyevolve.sourceforge.net.

[7] D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401:788–791, 1999.

[8] D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In NIPS, volume 13, page 556C562, 2001.

[9] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from flickr tags. In SIGIR, 2007.

[10] X. Xu and Z. Niu. Automatic document tagging in social semantic digital library. In ICONIP, volume 2, pages 344–351, 2009.

[11] Q. Zhao, P. Mitra, and B. Chen. Temporal and information flow based event detection from social text streams. In AAAI, 2007.

Yingxing Li was born in Jilin, China in March., 1963. He got doctoral degree in school of biological and agricultural engineering, Jilin University, Chuang Chun, China in 2006. His research filed is focus on optimization of information system and management information system.

He is a professor at Beihua University, Jilin, China. He also is the dean of Province Research Institution of Regional Economy.

Ning Li was born in Shandong, China in Jan., 1977. She got doctoral degree in school of biological and agricultural engineering, Jilin University, Chuang Chun, China in 2008. Her research filed is focus on agricultural mechanical engineering and system engineering.

She is a lecturer at China university of petroleum, dongying, China.

