2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Query Space Exploration Using Genetic Algorithm

Jae-Hoon Lee* •Young-Cheon Kim** •Sung-Joo Lee***
* Dept. of Computer Science, Graduate School, Chosun Univ.
** Dept. of Information & Communication, Seojeong College.
*** Dept. of Computer Engineering, Chosun Univ.
E-mail : nuridepo@stmail.chosun.ac.kr

Abstract - **Information retrieval must be able to search the most suitable document that user need from document set. If foretell document adaptedness by similarity degree about QL(Query Language) of document, documents that search person does not require are searched. In this paper, showed that can search the most suitable document on user's request searching document of the whole space using genetic algorithm and used knowledge-base operator to solve various model's problem.**

Ⅰ. Introduction

Fast growth of internet is spreading fast by different industry and effort is required so that can do suitable information retrieval. Information retrieval models such as Boolean, Probabilistic, Vector space, Adaptedness feedback model had been studied interval long.

Boolean model is based on Set Theory and Boolean Algebra and offers operator of *and*, *or*, *not* and so on. This model is advantage that can offer clear Formalism and express meaning that user wants exactly. But, because partial matching is impossible, overfull document is searched or there is shortcoming that can not do sequencing because only minimum's document can be searched and being in difficulty degree with query is decided by 0 or 1.

Vector space model can give proper weight(not binary) on QL or keyword of document and partial matching is possible. Therefore, can do document sequencing in similarity degree of query and document.

Probabilistic model defines relationship and dependency between words, weight of query, similarity degree between query and document in this model. While probability model has advantage that can define model to do formal and decide all parameters in this model, but have shortcoming that must presume early parameter.

Lately, interest was augmented to mechanical studying method. Specially, is expecting in natural selection processing by analogical inference using Neuron and evolution algorithm.

Gene algorithm developed by Holland and is basing on Darwin's survival theory and nature evolution processing principle. Gene algorithm can improve formulation of QL, and is offering domain of information retrieval in optimized document.

This paper showed that can search for the most suitable document gathering on user's request searching several parts of document space using gene algorithm.

Ⅱ. Query Space Exploration Using Genetic Algorithm

2.1 Individual and Population

At gene algorithm, Individual mean query and gene or chromosome is coincided index word or concept. Early locus is word weight, and early population is consisted of early question and searched suitable document list. Query of each individual appear as following.

$$Q_u(q_{u1}, q_{u2}, \cdots q_{ut}) \quad (1)$$

Query weight(q_{ui}) is calculated by query word weight scheme. Scheme is subset of all possible strings that have same bit value in decided string position. Query weight evolves through generation. Query weight formula is as following.

$$q_{ui} = \frac{(1+\log(tf_{ui}))\times(\log\frac{N}{n_i})}{\sqrt{\sum_{k=1}^{T}((1+\log(tf_{nk}))\times(\log(\frac{N}{n_k}))^2}} \quad (2)$$

T displays whole move of word drawn automatically from document and N display whole move of document. tf_{ui} expresses ti's frequency in document u. Early population $Pop(0)$ is consisted of early query and suitable document search list by early query. Find suitable document from query work to do first time for gene algorithm execution, and early population is not Random structure. Gene algorithm makes inquiries by good perfume from document space. Population is created newly after it repeats gene algorithm.

2.2 Fitness function

Fitness is calculated according to adaptedness of document that is given in each query of population and is searched. Fitness is proportional in near degree on documents that function is suitable and is inverse proportion in near degree on amiss documents. Fitness function formula is as following.

$$QFitness(Q_u^{(s)}) = \frac{\frac{1}{\|D_r\|}*\sum_{dj\in Dr} Sim(d_j,Q_u^{(s)})}{\frac{1}{\|D_{nr}\|}*\sum_{dj\in Dnr} Sim(d_j,Q_u^{(s)})} \quad (3)$$

D_r means gathering of suitable document search in all generation, and D_{nr} means set of document search bandit conjunction in all generation.

According to Jaccard function, Similarity value ($Sim(d_j,Q_u^{(s)})$) is as following.

$$Sim(d_j,Q_u^{(s)}) = \frac{\sum_{i=1}^{T} q_{ui}^{(s)} \bullet d_{ji}}{\sum_{i=1}^{T} q_{ui}^2 + \sum_{i=1}^{T} d_{ui}^2 - \sum_{i=1}^{T} q_{ni}^{(s)} \bullet d_{ji}} \quad (4)$$

Niche's definition searches document of resembling inherited character from query set. Niche's formula improves doing to search different document space evenly decoration, as following.

$$Niche(Q_u^{(s)}) = \{Q_u^{(s)} as\ ED(Q_u^{(s)},Q_v^{(s)}) \le \text{critcal value}\} \quad (5)$$

Euclid distance, $ED(Q_u^{(s)},Q_v^{(s)})$, is as following.

$$ED(Q_u^{(s)},Q_v^{(s)}) = \sqrt{\sum(q_{ui}^{(s)} - q_{vi}^{(s)})^2} \quad (6)$$

Niche's critcal value is decided by underground method. Niche's weight function way is as following.

$$Fitness(Q_u^{(s)}) = \frac{QFitness(Q_u^{(s)})}{\|Niche(Q_u^{(s)}\|} \quad (7)$$

Query Fitness is proportional in query similarity degree of suitable document in same Niche's query number and Document bandit conjunction is inverse

proportional in query similarity degree. New domain is examined according to average fitness connected with query. Query population, $Pop^{(s)}$, is composed as following.

$$Pop^{(s)} = \{(Q_u^{(s)}, Fitness(Q_u^{(s)})), \mathrm{u} = 1,2,\cdots \mathrm{Popsize}\} \quad (8)$$

2.3 Gene operator

In GA, kind of operator is Selection, Crossover, Mutation. In this paper, we use two Knowledge-based Crossover Operator(Term Weight Based, Term Co-occurrence Based) and Blind Crossover Operator.

2.3.1 Selection

Selection selects chromosome of population to two for hybridization and use roulette wheel selection traditionally. After yield ratio with relative value of conjunction of fidelity of all chromosomes and fidelity of each chromosome in population, chromosome that have of dominant character is existed more in next generation averagely because have as selected possibility as the probability and recessive chromosome is selected.

2.3.2 Crossover

Crossover consists of pair of individual that is selected by crossover probability value. Use two intersection points selected random to create chromosome of next generation. Select position to contact according to predetermined rules after copy two chromosomes that is selected by selector and change part of chromosome each other. This time sphere that change is localized on parameter interior of selected position. After crossover, genetic factors of parent chromosome create confidence chromosome that have new argument being composed.

Crossover that do fetters in word weight uses query belonging to same Niche and that do fetters to word co-occurrence uses query from other Niche.

2.3.3 Mutation

Mutation can prevent that fall in part optimization by select argument of dyeing interior of the body unintentionally and act role that create new argument in chromosome because exchanging the cost to love and recover genetic factor that lose at evolution front step. This paper uses two mutation operators.

① Mutation based word adaptedness

Change weight of specification word by weight mean value of all words in query mutation here to achieve mutation by fixed probability to words that show on query.

$$score(t_i) = \frac{\sum_{dj \in Dr^{(s)}} d_{ji}}{\| D_r^{(s)} \|} \quad (9)$$

② Blind Mutation

This operator changes weight cost by fixed probability by random value to words that show on query.

Ⅲ. Experiment and Result

Estimation method measured accuracy augmenters of high position 15 document through repeat of gene algorithm in this treatise. That repeat much is to search the most suitable document and estimate document order. In an experiment, Repeat fixed by 5 times and whenever repeat creates new generation through new gene algorithm. At this process, decide adaptedness about high position 15 documents that make new document list and is decided that is suitable. The goal of gene algorithm is searching for new adaptedness document through repeat

of search. Method to measure efficiency of gene algorithm is shown in **Algorithm 1**.

Table 1. Method to measure efficiency of gene algorithm

1. About early query search achievement.
2. Adaptedness decisions about high position 15 document.
3. Early population creation.
   3.1 Add decided documents to population($Pop^{(s)}$) that is suitable with early query.
   3.2 Adaptedness of each individual decision.
   3.3 Gene operator application.
   Repeat
   ① Select two parents by wheel-selection
   ② Crossover two parents by specification probability.
   ③ Mutate created descendants by specification probability.
   Until reach in size of population, repeat.
4. Achieve search about each query in population.
5. Search result adding up.
6. Adaptedness decisions about high position 15 document.
7. Calculate Fitness of each query.
8. Gene algorithm achievement.
9. Repeat 4~8.

In Table 2., gene algorithm using operator of G1(based on term weight) and G2(based on term co-occurrence) to measure parameter crossover rate(P_c), mutation rate(P_m), population size performance experiment.

Table 2. An experiment by crossover rate(P_c), mutation rate(P_m).

Rept.	1	2	3	4	5
P_m	P_c=0.1				
0.01	91(91)	36(127)	43(170)	52(223)	36(259)
0.07	91(91)	36(127)	43(170)	52(223)	37(260)
0.1	91(91)	36(127)	43(170)	52(223)	37(260)
0.25	91(91)	36(127)	43(171)	52(224)	35(259)
0.5	91(91)	36(128)	43(175)	49(224)	36(260)
P_m	P_c=0.25				
0.01	89(89)	60(149)	43(192)	36(228)	41(269)
0.07	89(89)	60(149)	43(192)	36(228)	40(268)
0.1	89(89)	60(149)	43(192)	36(228)	40(268)
0.25	89(89)	60(149)	43(192)	36(228)	40(268)
0.5	89(89)	60(149)	43(192)	36(228)	38(268)
P_m	P_c=0.5				
0.01	92(92)	49(141)	49(190)	48(238)	37(275)
0.07	92(92)	49(141)	49(190)	48(238)	36(274)
0.1	92(92)	49(141)	49(190)	48(238)	36(274)
0.25	92(92)	49(141)	49(190)	48(238)	37(275)
0.5	92(92)	49(141)	47(190)	46(234)	34(266)
P_m	P_c=0.7				
0.01	94(94)	54(148)	69(217)	39(256)	36(292)
0.07	94(94)	54(148)	69(217)	39(256)	36(292)
0.1	94(94)	54(148)	69(217)	39(256)	36(292)
0.25	94(94)	54(148)	69(217)	39(256)	34(290)
0.5	94(94)	54(148)	68(216)	40(256)	34(290)

Table 2. experiment sets $P_c \in \{0.1, 0.25, 0.5, 0.7\}$), $P_m \in \{0.01, 0.07, 0.1, 0.23, 0.5\}$), size of population by 5. Value in parenthesis is accumulated whole summation of suitable document search. Experimented these process 5 times repeatedly, and repeat 0 is documents that is searched by early query. P_c and P_m act important role in information retrieval of gene algorithm.

If see in Table 2., can know that is influenced in crossover rate more than the mutation rate. If examine accumulated whole sum, displaying 259 when is =0.1, and is displaying 292 when is =0.7. According to result, some point of doubt occurs. This is influenced on mutation operator. Mutation operator is causing very

small effect at gene algorithm, and use very small probability value(0.005). But, this operator is very important at algorithm. Firstly, it is to use traditional gene algorithm, and second is to recombine genetic factors efficiently after crossover operation. This can recover damage of important genetic factor through crossover operation effectively. Used =0.7, =0.07 in this paper.

Population size(*Popsize*) decides factor. Size of big population generates noise, but can execute many queries. Small size breeds silence because genetic factor transformation of reason occurs from same document space.

In Table 3, show experiment result and that unuse that use gene algorithm by size of population in operator use of G1(based on term weight) and G2(based on term co-occurrence). Lists of Table 3 are expressing value of suitable document search, and show document searched whole summation that value in parenthesis is suitable.

Table 4 is comparing each other using Knowledge-based operator method and Blind operator method after fix size of population by 6. Knowledge-based operator method is expressing efficiency of double almost more than Blind operator method.

Table 3. Comparison of G1 and G2 by size of population

	G1				
	iter1	iter2	iter3	iter4	iter5
Popsize =2					
N_GA	90(90)	48(138)	33(171)	46(217)	24(241)
Y_GA	72(72)	52(124)	36(160)	46(206)	24(230)
Popsize =4					
N_GA	90(90)	57(147)	50(197)	42(239)	42(281)
Y_GA	96(96)	64(160)	40(200)	45(245)	42(287)
Popsize =6					
N_GA	90(90)	57(147)	39(186)	44(230)	37(267)
Y_GA	96(96)	64(160)	55(215)	51(266)	30(296)
Popsize =8					
N_GA	90(90)	63(154)	42(196)	37(233)	42(275)
Y_GA	96(96)	72(168)	50(218)	38(256)	29(285)

	G2				
	iter1	iter2	iter3	iter4	iter5
Popsize =2					
N_GA	90(90)	59(149)	36(185)	40(195)	21(216)
Y_GA	72(72)	42(112)	42(154)	35(189)	21(210)
Popsize =4					
N_GA	90(90)	54(144)	46(190)	40(230)	41(273)
Y_GA	85(85)	40(125)	46(171)	58(229)	45(273)
Popsize =6					
N_GA	90(90)	64(154)	40(194)	31(225)	31(256)
Y_GA	85(85)	64(149)	41(190)	51(241)	37(278)
Popsize =8					
N_GA	90(90)	64(154)	36(190)	40(230)	30(260)
Y_GA	85(85)	64(159)	45(204)	36(240)	30(270)

Table 4. Comparison of Knowledge-based operator and Blind operator

operator	iter1	iter2	iter3	iter4	iter5
Knowledge-based					
G1	96(96)	64(160)	55(215)	51(266)	30(296)
G2	85(85)	64(149)	41(190)	51(241)	37(278)
Blind					
G1	90(90)	23(113)	23(136)	32(168)	22(190)
G2	27(27)	28(55)	41(96)	28(126)	24(150)

Table 5 is expressing value of suitable document

search in repeat of gene algorithm. Critical Value acts very important role in Niche operator. The best critical value appeared through 0.1 ~ 0.2.

Table 5. Niche result by critical value

Critical value	iter1	iter2	iter3	iter4	iter5	Total
0	85	64	41	51	37	278
0.01	105	56	51	51	41	304
0.25	94	60	51	50	34	300
0.5	94	60	48	50	42	294
1	88	64	54	42	48	296
2	87	69	63	34	29	282
5	86	75	60	30	31	282
8	85	84	42	29	34	274
10	85	86	61	47	25	283

In Table 6, compare G1, G2, G3, Niche's performance was estimated as is good. G3 is more effective in case component is discrete more than G1, G2 method. Use Niche technology in gene algorithm search, performance improved.

Table 6. Compare G1, G2, G3

	iter1	iter2	iter3	iter4	iter5
G1	96(96)	64(160)	55(215)	51(266)	30(296)
G2	85(85)	64(149)	41(190)	51(241)	37(278)
G3	105(105)	56(161)	51(212)	51(263)	41(304)

Ⅳ. Conclusion

In experiment result, certified that information retrieval that improve more if use gene algorithm in information retrieval system.

This paper implements and experimented following things. First, create query population newly. Second, it is able to search different sacred ground by probability from document space. Third, it is using Knowledge based operator not Blind operator that use by general gene algorithm. Fourth, it is used Niche technology to search documents that is discrete to different document space.

Experiment result depends on probability values of much parameter, size of population, critical value and produces result through gene algorithm. This paper shows in information retrieval how gene algorithm can be utilized. Gene algorithm is many problems in external form. Direct comparison did not with other Relevance Feedback method. But, through an experiment, GA can be used usefully to IR. Specially, capacitate estimate which may show good performance from document set that documents of mixing(heterogeneous) collect.

Reference

[1] Boughanem M and Soule-Dupuy, Query modification based on relevance backpropagation. In: Proceedings of the 5th International Conference on Computer Assisted Information Searching on Internet, Montreal, pp. 469-487, 1997.

[2] Chen Machine learning for information retrieval: Neural networks, symbolic learning and genetic algorithms. JASIS, 46(3):194-216, 1995.

[3] Davis Handbook of Genetic Algorithms. Van Nostram Reinhold, New York, 1991.

[4] Haines D and Croft WB Relevance feedback and inference networks. In: ACM SIGIR International Conference on Research and Development in Information Retrieval, pp.2-11, 1993.

[5] Gordon M Probabilistic and genetic algorithms for document retrieval. Communications of the ACM, pp. 1208-1218, 1988.

[6] Goldberg DE Algorithmes genetiques. Exploration, optimisation et apprentissage automatique. addison Wesley, France, 1994.

[7] Harmaqn D TREC overview. In: 6th International

Conference on Text Retrieval TREC6, November 21-23. Harman DK,ed. NIST SP, PP. 1-24, 1997.

[8] Robertson S and Walker S, On relevance weights with little relevance information. ACM/SIGIR International Conference on Research and development in Information Retrieval, pp. 16-24, 1997.

[9] SEbag M and Schoenauer M, Controle un algorithme genetique. Revued' intelligence artificielle, 2/3:389-428, 1996.

[10] Yang JJ and Korhage R, Query optimization in information retrieval using genetic algorithms. ICGA, 1993.