

인덱스 부호 최적화를 위한 Extended Least Difference Greedy Clique-cover 알고리즘의 개선

이준우*, 노중선*

*서울대학교

*joonwoo3511@ccl.snu.ac.kr, *jsno@snu.ac.kr

Improvement of Extended Least Difference Greedy Clique-cover Algorithm for Index Code Optimization

Joonwoo Lee*, Jong-Seon No*

*Seoul National Univ.

요약

본 논문에서는 인덱스 부호의 길이를 최소화하기 위해 Kwak 등의 논문에서 제시한 발견적 알고리즘인 Extended Least Difference Greedy Clique-cover(ELDG) 알고리즘을 더 효과적인 결과를 내도록 수정하였다. ELDG 알고리즘에서 중요한 부분 중 하나인 column-merging heuristic 알고리즘이 인덱스 부호의 길이를 최소값에 가까운 값으로 줄이는데 효과적이지 않을 뿐 아니라 side information이 많을 경우 그의 전신 알고리즘인 LDG 알고리즘의 결과 값보다 더 큰 길이를 출력한다는 것을 실험적으로 발견하였다. 본 논문에서 수정한 결과, column-merging heuristic 알고리즘은 인덱스 부호 길이를 효과적으로 줄이는 알고리즘이 되었음을 밝혔다.

I. 서론

TV, 영화, 인터넷 신문 등과 같이 하나의 송신자가 많은 수신자들에게 공통된 정보를 보낼 때 사용하는 채널인 방송 채널에서 각 수신자들에게 서로 다른 정보를 보내야 하는 경우가 종종 발생하는데, 이러한 상황을 모델링 한 문제가 인덱스 부호 문제이다.

일반적인 상황에서 가장 작은 크기의 인덱스 부호를 찾는 것은 NP-hard 문제이기 때문에 최적의 길이에 가장 가까운 부호를 찾는 것이 실제적으로 중요한 문제가 된다. 본 논문에서는 최적화된 인덱스 부호의 근사값을 효과적으로 찾는 알고리즘을 다룬다.

본 논문의 구성은 다음과 같다. II에서는 인덱스 부호의 정의와 중요한 사실을 다룬다. III에서는 최적화된 인덱스 부호의 근사값을 찾는 알고리즘으로 제안된 Extended Least Difference Greedy Clique-cover(ELDG) 알고리즘을 소개한다. IV에서는 ELDG 알고리즘의 한계점과 그의 개선방안을 제시하고, 이를 뒷받침하기 위한 실험 결과를 제시한다. V에서는 논문의 결론을 맺는다.

II. 인덱스 부호

방송 채널에서는 각 수신자에게 서로 다른 정보를 보내지 못하고 모든 수신자에게 공통된 정보를 보내야 하기 때문에 송신자는 채널을 통해 보내는 정보에 모든 수신자가 받아야 할 정보들을 실어야 한다. 이 때, 각 수신자가 이미 가지고 있는 정보를 송신자가 아는 경우에는 그러한 부가 정보를 이용하여 송신자가 보내야 하는 정보의 크기를 압축할 수 있다. 이렇게 줄여서 만든 부호를 인덱스 부호라고 한다.

인덱스 부호 문제의 핵심은 주어진 부가 정보에 대해 인덱스 부호의 최소 길이, 최소 길이의 인덱스 부호를 생성할 수 있는 인코딩 함수, 디코딩 함수를 찾는 데에 있다. Bar-Yossef와 Birk는[1] 주어진 부가 정보에 대해 Fitting matrix와 minrank라는 수학적 개념을 정의하여 선형 인덱스 부호의 최소 길이를 수식으로 나타내었다. 노드가 n 개인 directed graph

G 가 self-loop이 없을 때 $n \times n$ GF(2) 행렬 A 에 대해 $A_{ii} = 1 (1 \leq i \leq n)$ 이고 $(i, j) \in E$ 인 모든 $i, j (i \neq j)$ 에 대해 $A_{ij} = 0$ 을 만족하면 A 가 G 에 fit하다고 정의한다. 그리고 $\text{minrk}_2(G) := \min\{\text{rank}_2(A) | A \text{ fits } G\}$ 이라 정의한다. 이 때, 각 수신자들이 서로의 정보를 아는 상황을 나타낸 그래프인 side information graph G 에 대해, 선형 인덱스 부호의 최소 길이는 $\text{minrk}_2(G)$ 임을 Ziv Bar-Yossef 외 4명이 밝혔다. 그러나 $\text{minrk}_2(G)$ 를 구하는 문제는 NP-hard 문제이기 때문에 정확한 값을 구하는 것보다 시간 복잡도가 다항 함수인 알고리즘을 사용하여 길이가 최소 길이에 최대한 가까운 근사값을 내는 것이 중요하다.

III. Extended Least Difference Greedy Clique-cover(ELDG) 알고리즘

ELDG 알고리즘은 Least difference greedy clique-cover(LDG) 알고리즘을 개선한 알고리즘이다.[2] LDG 알고리즘은 Birk와 Kol이 최적화된 인덱스 부호의 길이의 근사값을 효과적으로 얻기 위해 고안한 알고리즘이다.

LDG 알고리즘에서는 먼저 G 에 fit될 수 있는 행렬을 0, 1, * 세 개의 원소로 이루어진 행렬로 나타낸다. * 원소는 아직 결정되지 않은 원소에 위치한다. 그리고 * 원소에 0 혹은 1로 채우려고 한다. 두 행을 같게 하려고 할 때 채워 넣어야 하는 *가 최소 개수를 갖는 두 행을 골라 같게 하는 과정을 어떤 두 행도 같게 할 수 없을 때까지 계속 반복한다. 이러한 과정을 진행하기 위해 행간의 거리를 정의한다. 먼저 원소들 사이의 거리를 다음과 같이 정의한다.

$$d(0,0) = d(1,1) = d(*,*) = 0$$

$$d(0,*) = d(1,*) = 1$$

$$d(0,1) = \infty$$

그리고 두 행의 대응되는 원소들의 거리를 모두 더하여 행간의 거리를 정의한다. 대응할 수 있는 모든 두 개의 행들에 대해 거리를 계산한 후에

가장 작은 거리를 가지는 두 행을 골라 같게 하는 과정을 계속 반복한다. 모든 대응할 수 있는 두 행의 거리를 계산한 값이 모두 ∞ 가 나오면 알고리즘을 멈춘다.

ELDГ 알고리즘에서는 LDГ 알고리즘에 column-merging heuristic 알고리즘과 cycle-of-three-nodes detection 알고리즘을 추가했다. Column-merging heuristic 알고리즘에서는 LDГ 알고리즘을 변형하여 행간의 거리를 계산할 뿐 아니라 열간의 거리도 계산하여 가장 작은 거리를 가지는 것이 행간의 거리이면 두 행을 같게 하고, 가장 작은 거리를 가지는 것이 열간의 거리이면 두 열을 같게 하는 과정을 계속 반복한다. Cycle-of-three-nodes detection 알고리즘에서는 세 행을 더했을 때 모두 0이 나올 수 있도록 하는 데에 목적이 있다. 먼저 두 행의 덧셈을 정의하는데, 그를 위해 다음과 같이 원소들의 덧셈을 정의한다. 두 개의 피연산자 중 하나라도 *이 있다면 *를 결과로 한다. 그렇지 않으면 단순히 덧셈을 한다. 이렇게 정의한 덧셈을 사용하여 두 행의 대응되는 원소들끼리 더하여 새로운 행을 만든다. 이렇게 더한 행과 같게 할 수 있는 행을 찾기 위해 더한 두 행과 다른 나머지 행들과의 행간 거리를 계산하여 모두 ∞ 이면 그대로 두고, 그렇지 않으면 가장 짧은 거리를 갖는 행을 골라 세 개의 행을 더했을 때 0이 되도록 하기 위해 *에 원소들을 채운다.

ELDГ 알고리즘을 사용하여 줄인 코드의 길이가 LDГ 알고리즘을 사용하여 줄인 코드의 길이에 비해 평균적으로 10% 줄어든 효과를 보였다고 Kwak 외의 논문에서 확인하였다.[2]

IV. ELDГ 알고리즘의 개선

본 논문에서는 ELDГ 알고리즘의 column-merging heuristic 알고리즘이 LDГ 알고리즘을 개선하는 효과를 보이지 않는다는 것을 밝혔다. 본 논문에서는 column-merging heuristic 알고리즘의 효과를 알기 위하여 시뮬레이션을 통해 실험을 하였다. 시뮬레이션을 위해 C++ 언어를 사용하였다. 각 실험에서는 fitting matrix를 무작위로 생성되되, 대각 원소를 제외한 원소들에 0이나 *를 일정한 확률로 넣었다. *를 넣는 확률을 p 라 할 때 p 를 0부터 1까지 0.1씩 올리면서 실험을 하였다. 각 p 값에 대해 100개의 fitting matrix를 생성하여 알고리즘을 거친 후의 rank의 값을 계산하여 평균을 취하였다. 노드의 개수는 40개로 실험하였다.

[그림1]은 실험 결과를 그래프로 나타낸 것이다. x 축은 *를 넣는 확률을 의미하고 y 축은 기존 LDГ 알고리즘으로 도출된 인덱스 코드 길이에 비해 줄어든 정도를 백분율로 나타낸 것이다. 여기서 y 값이 양수라는 의미는 줄어들었다는 의미이다. 실험 결과, $0 < p < 0.5$ 의 범위에 있을 때에는 기존 LDГ 알고리즘의 결과와의 유의한 차이가 나타나지 않고, $0.5 < p < 1$ 의 범위에 있을 때에는 오히려 LDГ 알고리즘을 통하여 계산된 코드의 길이에 비해 코드의 길이가 큰 폭으로 증가하여 최대 14.3%만큼 증가하였다. 이를 통하여 column-merging heuristic 알고리즘은, ELDГ 알고리즘이 LDГ 알고리즘보다 좋은 성능을 가지게 되는데 효과적인 기여를 하지 못한다는 것을 실험적으로 알 수 있었다.

column-merging heuristic 알고리즘에서는 행간 거리만을 비교하는 LDГ와는 달리, 행간 거리와 열간 거리 모두를 보기 때문에 모든 거리가 무한대가 될 때까지 줄일 수 있는 횟수는 평균적으로 높다는 점에서, Kwak 외의 논문에서는 LDГ 알고리즘보다 rank를 줄일 수 있을 것이라고 주장하고 있다. 그러나 본 실험을 통하여 거리를 줄이는 횟수를 최대화해도 rank 값을 최소로 만들지 못할 뿐 아니라 서로의 정보를 많이 알면 알수록 기존 행간 거리만 비교할 때보다 성능이 더 저하됨을 알 수 있다.

따라서 이 column-merging heuristic 알고리즘을 수정하였다. 본 논문

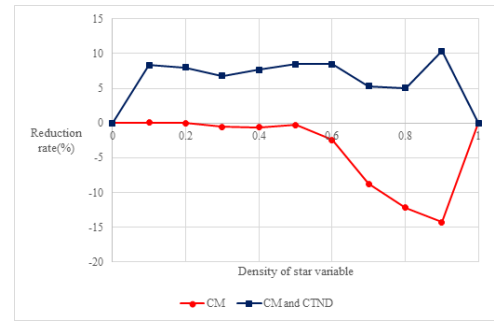


그림 1 기존 ELDГ 알고리즘의 분석

에서 수정한 알고리즘(new column-merging heuristic)에서는 총 줄일 수 있는 횟수에 주목하지 않았다. 행간 거리만을 비교하여 최대한 줄이고, 독립적으로 열간 거리만을 비교하여 최대한 줄인 다음, 두 가지 결과를 비교하여 더 적은 rank를 내는 값을 취하였다.

결과적으로, [그림 2]에서 볼 수 있듯이, 이를 다 합친 최종 결과 값 감소율은 기존 ELDГ 알고리즘에 의한 결과 값의 감소율(LDГ 기준)보다 최대 20.1% 포인트, 평균적으로 6.7% 포인트 커졌다는 것을 알 수 있다. 특히 *의 값이 많을 때에도 정상적으로 LDГ 알고리즘의 성능을 더 높이고 있음을 알 수 있다. 종합적으로는 기존 ELDГ 알고리즘에 의한 결과 값에 비해 대략 6.9% 감소한 결과를 보였다.

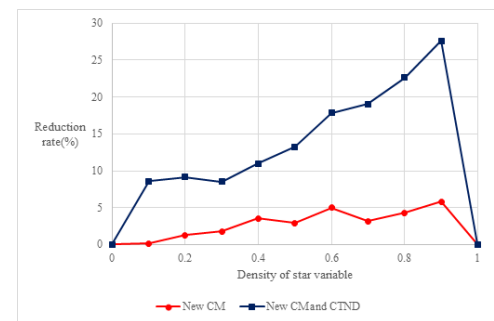


그림 2 새로운 Column-merging heuristic 알고리즘을 추가한 후 분석

V. 결론

이로써 다음의 결과들을 통하여 ELDГ 알고리즘의 column-merging heuristic 알고리즘을 수정하였을 때 기존 알고리즘과는 달리 유효한 결과가 나타나게 된다는 것을 알게 되었다. 결론적으로 새로운 ELDГ 알고리즘은 기존 ELDГ 알고리즘에 비해 평균적으로 대략 6.9% 정도의 코드 길이 감소의 효과를 보임을 알 수 있었다. 수정된 ELDГ 알고리즘이 더 높은 성능을 보임을 실험적으로 증명해내었지만, 수학적으로 어떤 원리로 인하여 성능이 높아질 수 있는지 제시하지는 못했다. 따라서 더 명확한 증명을 위해서 알고리즘에 대한 이론적인 분석이 진행되어야 할 것이다.

참고 문헌

- [1] Z. Bar-Yossef, Y. Birk, Index Coding With Side Information, IEEE Transactions on Information Theory, Vol. 57, No. 3, 1479, March 2011
- [2] S. Kwak, J. So and Y. Sung, An Extended Least Difference Greedy Clique-Cover Algorithm for Index Coding, IEEE International Symposium on Information Theory, 2014