웹 검색을 위한 준-지도방식 랭킹 학습

김계현○ 최승진
포항공과대학교
fenrir@postech.ac.kr, seungjin@postech.ac.kr

Scalable Semi-supervised Preference Learning for Web Search

Kye-Hyeon Kim○ Seungjin Choi
Pohang University of Science and Technology (POSTECH)

1. 서론

본 논문은 웹 검색에서 사용자의 검색 기록과 웹 문서간의 연관 관계를 동시에 이용하여 적합한 랭킹 함수를 학습하는 방법을 소개한다. 웹 검색 환경에서 대다수 사용자들은 최상위 10건 내외의 검색 결과만 확인하기 때문에, 11위 이하의 검색 결과들 중에서 실제로 질의어와 관련성이 매우 높은 웹 페이지들이 존재하는 경우, 일반적인 선호도 학습 방법으로는 이들의 순위를 올바르게 보정하기가 어렵다. 본 논문에서는 이러한 한계를 해결하기 위해 기존의 선호도 학습 문제를 준지도 학습(semi-supervised learning)으로 확장하였다. 준지도 학습이란, 수집된 전체 데이터 중 명시적인 정보를 가진 아이템(labeled data)의 양이 부족한 경우, 명시적인 정보가 없는 나머지 아이템(unlabeled data)도 학습에 함께 활용하여 성능을 높이고자 하는 학습 방법이다. 웹 검색의 경우, 수집된 전체 웹 페이지들 중 명시적인 정보를 가진 아이템(즉, 검색 기록을 통해 선호도 정보를 얻을 수 있는 웹 페이지)들은 질의어당 상위 10개 내외에 불과하므로, 준지도 학습이 필요한 적절한 예라고 할 수 있다.

제안하는 방법은 그래프 기반의 준지도 학습(semi-supervised learning) 기법을 선호도 학습(preference learning)에 적용한 기계학습 알고리즘으로, 정보 검색 분야에서는 최근까지 이와 유사한 여러 방법들이 이미 제안된 바가 있다. 하지만 기존의 연구들은 scalability를 전혀 고려하지 않아, 웹 검색과 같이 매우 규모가 큰 정보 검색 문제에는 사용하기가 불가능하다. 예를 들어, 기존 연구 중 가장 scalability가 나은 편인 최신 방법[1]의 경우, 불과 수만 개의 문서에 대해 매 질의어당 실시간으로 요구되는 학습 시간이 무려 100~200초에 이른다.

2. 본론

본 논문에서는 준지도 학습의 scalability 문제를 해결하는 근사(approximation) 알고리즘을 제안한다. 우선 그래프의 가중치 행렬(weight matrix)을 직접적으로 계산할 필요가 없는 matrix-free 알고리즘을 고안하여 대규모 데이터를 다룰 수 있도록 하였으며, 또한 새로운 검색 기록들이 추가될 때마다 이미 학습된 랭킹 함수를 효율적으로 업데이트할 수 있도록 점진적(incremental) 학습 알고리즘을 개발하였다. 알고리즘의 전체적인 구성은 다음과 같다.

가장 먼저, 주어진 검색 기록 목록에서 중복된 질의어를 제거하여 총 M개의 서로 다른 질의어를 추출해내는 과정이 필요하다. 본 논문의 실험에서 사용한 MSN Live Search 데이터의 경우, 총 1225만여 개의 검색 기록 중 387만여 개의 서로 다른 질의어를 추출하였다. 다음으로, 상용 검색 엔진(본 논문의 실험에서는 Live Search API를 이용)을 이용하여 각 질의어당 상위 K개의 웹 페이지 URL을 수집한다(본 논문에서는 K=20). 하나의 웹 페이지가 여러 질의어의 top-K에 속할 수 있으므로, 이 과정에서 총 M*T개 이하의 URL을 수집하게 된다. 이 URL들이 바로 학습에 사용할 전체 아이템 집합이 된다(즉 N≤MT). 다음으로 수집된 URL들의 유사도를 다음과 같이 정의한다.

$$[\boldsymbol{W}]_{ij} = \frac{\boldsymbol{x}_i^\top \boldsymbol{x}_j}{||\boldsymbol{x}_i|| \cdot ||\boldsymbol{x}_j||} \quad \cdots\cdots\cdots\cdots \quad (1)$$

이때 $\boldsymbol{x}_i$ 는 M차원 벡터를 나타내며, i번째 URL이 k번째 질의어에 대해 top-K에 속하는 경우 $[\boldsymbol{x}_i]_k = 1$, 속하지 않을 경우 $[\boldsymbol{x}_i]_k = 0$으로 정의한 벡터이다. 따라서 벡터의 크기(norm) $||\boldsymbol{x}_i||$는 i번째 URL을 top-K로 가지는 질의어들의 개수를 의미하며, $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ 는 i번째 URL과 j번째 URL을 함께

top-K로 가지는 질의어들의 개수를 의미한다. 어떤 질의어에 대해 두 웹 페이지가 상위 결과에 함께 오른다는 것은, 해당 질의어의 관점에서 두 웹 페이지의 내용이 서로 유사함을 뜻한다. 만약 이러한 경우가 여러 질의어에 대해 일어났다면, 그만큼 두 웹 페이지는 보편적으로 유사한 내용을 가지고 있다고 할 수 있다. 본 논문에서는 이와 같이 정의된 유사도를 이용하여 준지도 학습 문제를 풀었다. 수집된 웹 페이지들의 함수 값들을 $\boldsymbol{f} = [f(x_1), f(x_2), \dots, f(x_N)]^{\mathsf{T}}$ 와 같이 벡터로 표현하면, $\boldsymbol{f}$의 rank-R 근사치는 $\boldsymbol{W}$의 R차원 주성분 공간(principal component space)상에 존재하게 된다. 다시 말하면, $\boldsymbol{W}$의 R차원 주성분 공간을 $\boldsymbol{v}_1, \dots, \boldsymbol{v}_R$ (각각 N차원 벡터) 이라고 했을 때, 다음의 등식을 rank-R 근사 오류 이하로 만족하는 벡터 $\boldsymbol{u} = [u_1, \dots, u_R]^{\mathsf{T}}$이 임의의 $f(x_i)$에 대해 항상 존재한다.

$$f(x_i) \approx \sum_{r=1}^{R} u_r [\boldsymbol{v}_r]_i \quad \dots\dots\dots (2)$$

본 논문에서는 이러한 성질을 이용, N보다 훨씬 작은 값 R에 대해(본 논문에서는 R=200) 위의 등식을 만족하는 벡터 $\boldsymbol{u}$와 $\boldsymbol{W}$의 R차원 주성분 공간 $[\boldsymbol{v}_1, \dots, \boldsymbol{v}_R]$을 구하였다. 우선 주성분 공간, 즉 top-R Eigenvector는 N이 매우 큰 경우에 적합한 power iteration을 사용하였는데, 본 논문에서는 power iteration에서 가장 핵심이 되는 부분인 '$\boldsymbol{W}$와 $\boldsymbol{v}_i$의 곱'을 $\boldsymbol{W}$ 없이 계산할 수 있는 알고리즘을 다음과 같이 유도하여(식 (1)의 유사도 정의를 이용), $\boldsymbol{W}$를 유지하는 데에 필요한 $O(N^2)$의 공간을 절감하였다.

$$[\boldsymbol{W}\boldsymbol{v}]_i = \sum_{[\boldsymbol{W}]_{ij} > 0} [\boldsymbol{W}]_{ij} [\boldsymbol{v}]_j = \sum_{\substack{c: \\ Q_{ci}=1}} \sum_{\substack{j: \\ Q_{cj}=1}} \frac{[\boldsymbol{v}]_j}{||\boldsymbol{x}_i|| \cdot ||\boldsymbol{x}_j||} \quad \dots. (3)$$

위 식에서 Q_{ci} 란, c번째 질의어의 top-K 검색 결과 중에 i번째 URL이 있는지 여부를 나타낸다(있으면 1, 없으면 0). 위 식을 바탕으로 수행되는 matrix-free power iteration 알고리즘의 시간 복잡도는 $O(MT) = O(N)$이다. 이와 같이 주성분 공간 $[\boldsymbol{v}_1, \dots, \boldsymbol{v}_R]$을 구하고 나면, 주어진 선호도 데이터를 따르도록 식 (2)의 벡터 $\boldsymbol{u}$를 최적화하는 것은 선형 회귀분석(linear regression)으로 간단히 풀 수 있다. 본 논문에서는 선호도 정보가 미리 주어져있지 않다거나, 또는 실시간으로 새로운 선호도 정보가 계속해서 추가되는 경우를 위해(상용 검색 엔진에 적용하고자 하면 이런 상황을 처리할 수 있어야 한다), 아래와 같이 임의의 두 문서 *A*, *B*에 대한 새로운 선호도 정보가 들어올 때마다 점진적으로 $\boldsymbol{u}$를 업데이트하는 알고리즘을 유도하였다(분량 제한으로 인해 유도 과정은 생략).

$$\boldsymbol{u} \leftarrow \eta \boldsymbol{u} + y(\boldsymbol{z}'_A - \boldsymbol{z}'_B) \quad \dots\dots\dots (4)$$

상수 $\eta > 0$는 새로운 정보에 대한 편향성(bias)을 나타내며, $\eta < 1$인 경우 더 최근에 수집된 선호도 정보를 더 중요하게 고려하게 된다. y는 선호도가 문서 *A*의 선호도가 *B*보다 큰 경우 1, 반대의 경우 -1이다.

3. 실험 결과 및 결론

본 논문에서는 Microsoft Research Asia에서 약 400만개 질의어에 대해 수집한 MSN Live Search의 검색 기록 데이터에 제안된 방법을 적용하여, 상용 검색 엔진인 Live Search와 성능을 비교하였다. 'Live Search의 검색 결과'와 '제안된 방법으로 보정한 검색 결과'를 각각 구하고, 각 결과 내에서 '사용자가 실제로 방문한 웹 페이지의 평균 순위'를 각각 측정하였다. 그 결과, 사용자가 실제로 방문한 웹 페이지가 Live Search에서 11-20위로 낮게 책정된 경우, 제안한 방법에서는 3-12위로 약 8단계 향상된 순위를 기록하였다. 매 질의어당 사용자들이 실제로 훑어보는 검색 결과 건수가 평균 10건 내외임을 고려했을 때, 기존의 Live Search에서는 사용자에게 노출되지 못하던 유용한(사용자가 실제로 관심 있어할) 웹 페이지들이, 제안된 방법을 통해 순위가 향상되어 사용자들의 탐색 범위까지 들어오게 된 것이다. 또한 3.2GHz Pentium 4 CPU에서 제안된 방법의 학습 시간을 측정한 결과, 약 40시간의 전처리(한 번 수행하고 나면 반복할 필요 없음)를 마친 이후, 매 질의어당 실시간으로 소요된 처리 시간은 1.4밀리초에 불과했다. 이러한 전처리 및 실시간 학습은 2GB 메모리를 가진 PC에서 아무런 문제없이 모두 완료되었다. 기존의 최신 방법[1]에서의 처리 시간인 100~200초와 비교했을 때, sclability 면에서 큰 향상이 있었음을 확인할 수 있다.

상용 검색 엔진에서 수집된 실제 웹 검색 기록을 바탕으로 수행한 이러한 실험 결과는, 본 논문에서 제안한 방법이 실제 웹 검색에도 충분히 적용 가능한 scalability를 가지고 있으며, 단순히 이론적으로만 의미 있는 방법이 아니라, 실제로 상용 검색 엔진의 검색 정확도를 더욱 향상시킬 수 있는 방법임을 보여준다.

[1] K. Duh and K. Kirchhoff. Learning to rank with partially-labeled data. In Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval, 2008.