Term Weighting Schemes for Latent Dirichlet Allocation

Reference

- Term Weighting Schemes for Latent Dirichlet Allocation, Andrew et al.
- bab2min github

Introduction

기존의 LDA는 모든 단어를 동등하게 취급한다. 예를 들어 호텔 리뷰 데이터에서, corpus에 많이 등장하는 a, the와 같은 불용어나 service, food와 같은 핵심 단어를 동일하게 본다. 사실 컴퓨터 입장에서 이 단어들은 '빈도'에 의해서 판단될 뿐 a, the 같은 단어들이 불용어라는 것을 사전에 인지할 방법이없다. 따라서 불용어 처리를 적절하게 해주지 않으면 LDA의 주제 별 단어 분포에서 불용어가 고빈도 단어로 나올 가능성이높다. LDA가 모든 단어를 동등하게 보는 것은, LDA의 깁스 샘플링 식을 봐도 알 수 있다.

$$P(z_i \mid z_{-i}, w) \propto \frac{N_{n,k} + \beta}{\sum_n N_{n,k} + V\beta} \times \frac{N_{d,k} + \alpha}{\sum_d N_{d,k} + T\alpha} \tag{1}$$

깁스 샘플링의 단어가 나올 확률을 계산하는 식은, '빈도'에 비례함을 알 수 있다. 즉, 불용어나 핵심 단어나 많이 나오는 것을 구분하지 않는다는 것이다. 이 점에서 유추해볼 때, LDA를 하기 전에 불용어 처리를 잘 해야함을 알 수 있다.

하지만 어떤 것이 불용어인지 모르는, 생전 처음 보는 언어로 구성된 corpus에 topic modeling을 하는 경우라면? 또는 불용어를 일일이 확인하는 작업이 데이터의 갯수가 많아진다면 쉽지 않을 것이다. 이 논문에서는 weight을 사용하여, 이러한 작업을 자동화한다.

Weighting LDA

아이디어도 간단하고 깁스 샘플링에 적용하는 것도 간단하다. 위 (1)의 깁스 샘플링 식에 n번째 단어에 대한 weight인 w_n을 넣어주면 된다. (1)의 first term의 $N_{n,j}$는 n번째 단어가 k번째 주제에서 나온 빈도인데, 이미 word에 대한 summation이 있으므로 그대로 w_n을 써주면 된다. second term의 $N_{d,k}$는 d번째 문서가 k번째 주제에서 나온 빈도인데 여기에는 document에 대한 summation이 있으므로 따로 word에 대한 summation을 따로 만들어야한다. 이는 아래와 같다.

$$P(z_i \mid z_{-i}, w) \propto \frac{W_n N_{n,k} + \beta}{\sum_n W_n N_{n,k} + V\beta} \times \frac{W_n N_{n,d,k} + \alpha}{\sum_d \sum_n W_n N_{n,d,k} + T\alpha} \tag{2}$$

What Kind of Weight?

여기까지 살펴보면 어떤 가중치를 사용할지에 대한 질문이 자연스럽게 떠오른다. 이에 대해서 살펴보자.

- 1의 값으로 통일
  이는 기존의 LDA와 동일한 모형이 된다. 모두 동일한 가중치를 받기 때문이다.

- 0또는 1의 값
  이는 기존에 불용어를 처리해주는 작업을 하는 것과 동일하다. 불용어는 가중치 0을 받아서 아예 계산에서 제외하고 일반 단어는 1의 값을 주어서 계산에서 포함한다.

- 정보량 (혹은 idf)
  정보량은 해당 사건이 발생한 확률의 역수에 로그를 취한 값이다.

$$I = -log\, P(x) \approx log \frac{N}{df}$$

  여기서 N은 전체 단어가 발생한 빈도, df는 해당 단어가 발생한 빈도이다.

- 점별 상호 정보량 (PMI)
  정보량은 문서별로 동일한 가중치를 가정한다. 즉, food라는 단어가 각 문서에서 차지하는 중요도가 동일하다는 것이다. 하지만 food에서 중점적으로 얘기하는 문서와 곁다리로 food가 나오는 문서를 고려해볼 때, 각 문서에서 food가 차지하는 비중을 동일하다고 가정하는 것이 올바를까? 이러한 상황이 발생할 수 있기 때문에 문서별로 다른 가중치를 PMI를 통해 가정해본다.

$$PMI = log \frac{P(w \mid d)}{P(w)} = log \frac{N_{n,d} N}{N_d N_m}$$

  여기서 $N_{n,d}$는 n번째 단어가 d번째 문서에서 나타난 횟수, N은 전체 단어의 갯수, N_d는 d번째 문서 단어의 갯수, N_n은 n번째 단어의 전체 갯수이다.

Conclusion

흥미로운 점은, 이러한 weighting 트릭이 모든 topic modeling에 적용 가능하다는 점이다. 다양한 topic modeling이 연구되었으며 사후 분포에 대한 근사 분포를 유도하기 위해 variational bayes inference나 깁스 샘플링을 이용한다. 만약 어떤 topic modeling 논문이 깁스 샘플링을 이용해 사후 분포를 근사하였다면, weighting 기법을 써봐서 topic coherence을 비교하는 것도 괜찮을 것 같다.