커널 기반의 '단백질-단백질 작용' 의미 포함 문장 분류

김성환 ○ 엄재홍 장병탁
서울대학교 컴퓨터공학부
shkim@bi.snu.ac.kr○ jheom@bi.snu.ac.kr btzhang@bi.snu.ac.kr

Kernel-based sentence classification for protein-protein interaction

Seong-Hwan Kim ○ Jae-Hong Eom Byoung-Tak Zhang
School of Computer Science and Engineering, Seoul National University

요 약

본 논문에서는 tree kernel을 이용 '단백질-단백질 작용' 내용 포함 문장의 추출 방법을 제시한다. Tree kernel은 convolution kernel의 하나로서, 이를 이용하여 파싱 트리(parsing tree)로 표현된 문장을 데이터로 하여 '단백질-단백질 작용' 내용을 포함하고 있는 문장을 그렇기 않은 문장으로부터 분류할 수 있다. 문장 전체를 데이터로 사용하는 것보다 관련 영역을 서브트리(sub-tree)로 추출하여 사용한 것이 더 효과적임을 확인할 수 있었고, kernel 계산에 있어 파싱 트리의 태그 내용이 중요한 역할을 하기 때문에 이를 '단백질-단백질 작용'의 의미를 반영할 수 있도록 semantic하게 변환한 효과 및 트리의 길이에 따른 영향도 실험해 보았다. 문제에 사용된 데이터의 양이 다소 적었지만, 데이터 표현 방식에 따라 파싱이나 패턴 기법을 이용한 기존의 방법과 비교해 좋은 성능을 보일 수 있다는 가능성을 확인할 수 있었다.

1. 서 론

생물학적 기작 연구에 있어 단백질(단백질 생성 유전자 포함)간의 작용에 관한 내용은 주목받는 정보이다. 그동안 BIND와 같은 '단백질-단백질 작용'에 대한 내용을 저장하고 관련 정보를 제공하는 데이터베이스가 많이 생겨났지만 아직 많은 데이터가 관련 학술 문서상에 존재하고 있다. 따라서 문서상에 자연 언어(natural language)로서 존재하는 '단백질-단백질 작용'에 대한 정보를 자동으로 추출, 이용하기 위해 다양한 자연 언어 처리 기술(natural language processing technology)이 연구되었고 주로 파싱(parsing)기반의 방법[8]과 패턴 매칭(pattern matching) 이용한 방법[6]이 일반적으로 사용되어 왔다.

커널 기법(kernel method)은 고차원 특성 공간(feature space)으로의 매핑(mapping)과 특성 공간 내에서 두 데이터간의 내적 계산을 이용한 유사도 평가에 기초하고 있다. 커널 기법은 데이터 분류(classification) 문제에 효과적으로 사용되는 기계 학습(machine learning)방법으로 자연 언어 처리 분야에서도 BOW 커널(bag of word kernel)과 시퀀스 커널(sequence kernel) 등이 대표적으로 많이 사용되고 있다. [4, 7]

본 논문에서는 convolution 커널의 일종인 트리 커널(tree kernel)[5]을 이용하여 두 개의 단백질 내용을 포함하고 있는 문장이 '단백질-단백질 작용' 정보를 포함하고 있는지 아닌지를 판명하는 방법을 제시한다. 트리 커널은 두 데이터간의 공통 서브트리(sub-tree) 개수에 근거하여 두 데이터 간의 유사도를 측정하기 때문에 트리 형태로 표현된 데이터에 효과적으로 적용될 수 있다. 기존에는 parsing의 정확도[5]나 문장 내에서 predicate-argument 구조 파악 등 자연언어처리 분야의 일부 연구에 사용되었으나 다른 convolution kernel 기법에 비해 상대적으로 적용된 사례가 적은 편이다. 태깅(tagging)과 파싱(parsing)과정을 거쳐 문장을 트리 형태로 재구성하고 트리 커널 및 대표적인 선형 분류기(linear classifier) 모델의 하나인 SVM(support vector machine)를 이용하여 학습과 분류 성능 평가를 하였다.

2절에서는 커널 함수를 설명하고 3절에서는 트리 커널을 이용한 '단백질-단백질 작용' 정보 추출에 대해 설명한다. 4절에서는 실험에 사용된 예제 군의 특성, 파라미터 값 변형 및 태그 변형에 따른 실험 결과에 대해 비교 분석한 한다. 5절에서는 결론 및 향후 연구방향을 제시한다.

2. 커널 함수(kernel function)

일반적으로 문제 공간에서의 입력 데이터는 특정 속성들의 벡터(vector) 형태로서 표현되며, 이를 특정 가중치 벡터(weight vector) 및 임계값(threshold value)을 기준으로 분류할 수 있는 선형분류기(linear classifier) 모델[3]은 기계 학습에 있어 대표적인 분류 기법의 하나이다. 입력 데이터 공간(input space)상의 한 점으로 표현되는 입력 데이터를 다른 차원의 특성 공간상의 한 점으로 대응시키는 매핑(mapping) 함수를 이용, 특성 공간상에서 선형 분리가 가능한 형태의 데이터로 변환한다. 특성 공간상에서 선형 분류기 알고리즘(linear learning algorithm)을 적용 시, 특성 공간상에 표현된 두 데이터 간의 내적(inner product) 값을 계산해야 하며 이러한 일련의 과정을 외부적인 특성 공간 파악 없이 내부적으로 수행할 수 있는 것이 커널 함수(kernel function)이며 커널 함수와 선형 분류기 모델을 이용한 다양한 분류기법을 커널 기법이라 한다.

3. 트리 커널(tree kernel) 기반의 '단백질-단백질 작용' 정보 추출

3.1 '단백질-단백질 작용' 정보

본 논문에서 대상으로 하는 데이터는 문장을 구분 단위로 하는 텍스트 데이터이며 한 문장 내에서 '단백질-단백질 작용' 정보는 다양한 방식으로 나타난다. 정형화된 법칙은 없으나 'bind'나 'interact' 같은 단어를 통한 두 단백질 간의 직접적인 작용을 표현한 방식은 물론이고, 어느 한 단백질의 특정 작용이나 성질 혹은 부분 등이 대응되는 다른 것에 직접 혹은 간접적인 방식으로 영향을 미치는 등의 표현 방식 역시 생물학적인 지식에 근거한 판단을 통해 포함될 수 있다. 전자의 예로 "Further , we show that in a variety of cellular contexts , SGK phosphorylates CREB . " 와 같은 문장이 있을 수 있고 , 후자의 예로는 "We show that overexpression of hsTAF12 potentiates ATF7-induced transcriptional activation through direct interaction with ATF7 ." 와 같은 문장이 있다.

3.2 트리 커널(tree kernel)

텍스트 데이터 분류 문제에 있어 커널 기법이 적용된 사례가 많이 있다. 특정 단어의 출현 횟수를 속성 값으로 하는 벡터(vector) 형태로서 데이터를 표현하고 이러한 데이터의 내적 계산을 통해 커널함수를 구현한 BOW(Bag of Word) 커널이 대표적이다. [4, 7]

한편, 단순히 단어의 분포 특성을 이용한 데이터의 표현은 텍스트의 구조 정보를 반영하지 못하는 한계가 있으므로 텍스트 데이터의 구조적인 정보를 이용하여 커널을 구현하는 방법이 제시되었다. 텍스트 데이터를 문자의 시퀀스(sequence)로 보고 두 데이터간의 공통의 서브시퀀스(subsequence)를 속성으로 하여 그 횟수를 커널 함수 값으로 하는 시퀀스 커널(sequence kernel)이 대표적이며, 구조 정보를 recursive하게 계산하는 이러한 특성의 커널들을 convolution kernel이라 한다.

트리 커널(tree kernel) 역시 convolution kernel의 일종으로서 텍스트 데이터를 트리 구조로서 인식하는데 특히 부모 노드에서 파생된 자녀

노드에 대해 순서가 부여되는 구조로서 표현된다. 두 트리 데이터 간의 공통 서브트리(subtree)에 대해 전체 횟수를 합산한 것이 커널 함수 값이 된다. 따라서 트리 커널은 두 트리 간의 구조적인 유사도를 평가하는 효과적인 방법으로 사용될 수 있다.

고차원 특성 매핑(feature mapping) 과정을 통해 내부적으로 하나의 트리 데이터는 다음과 같이 해당 트리를 구성하고 있는 서브트리의 벡터(vector) 형태로 표현된다.

ϕ(Tree T) = (number of subtree of type 1, ... , number of subtree of type n)

이를 이용하여 구현한 커널 함수는 다음과 같다.

$$K(T_1, T_2) = \phi(T_1) \cdot \phi(T_2) = \sum_i \phi(T_1)[i] * \phi(T_2)[i]$$

$$= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) * I_i(n_2) \quad (1)$$

여기서 N_1과 N_2는 각각 트리 T_1과 T_2 내부의 모든 가능한 노드(node)들의 집합을 의미하며, $I_i(n)$은 타입 i의 서브트리가 n을 루트 노드(root node)로 시작된 경우 1, 그렇지 않은 경우 0 값을 갖는 indicator function이다.

트리 T내의 타입 i의 서브트리 개수는 $\phi(T)[i] = \sum_{n \in N} I_i(n)$이며, 타입 i의 서브트리를 갖는 트리 T내의 모든 노드의 개수를 의미한다.

두 트리 데이터 간의 모든 가능한 서브 트리를 속성으로 한 내적 계산은 다음과 같은 recursive한 방법을 통해 polytime 내에 계산 가능함 알려져 있다.

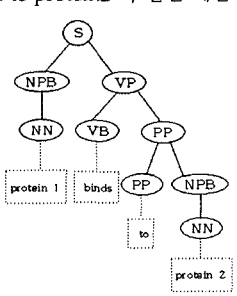
1) 만약 n_1, n_2 두 노드 간에 자녀 노드가 서로 다른 형태인 경우,
$NmComSubt(n_1, n_2) = 0$
2) 그렇지 않고 두 노드 간의 자녀가 동일한 형태이고(순서까지 포함) 두 노드가 leaves(POS tag)인 경우,
$NmComSubt(n_1, n_2) = \lambda$
3) 그 외의 모든 경우에 대해,
$NmComSubt(n_1, n_2) = \Pi_j(1 + NmComSubtr(n_1\ j\ , n_2\ j\ ))$

여기서, $NmComSubt(n_1, n_2) = \lambda \sum_i I_i(n_1) * I_i(n_2)$이다.

$\lambda(0 < \lambda \leq 1)$는 트리 조각(tree fragment)의 상대적인 중요도를 길이에 따라 반영하기 위해 적용하는 파라미터로서 이를 고려하지 않는 경우에는 1로 설정하여 사용한다.

3.3 텍스트 데이터의 파싱 트리표현

파싱 트리(**parsing tree**)는 트리 형태로서 텍스트 데이터의 내용 및 구조적인 정보를 함께 담고 있는 가장 대표적인 방법이다. 다음은 "protein1 binds to protein2"와 같은 예문을 파싱 트리로서 구현한 예이다.



3.4 전체 process

먼저 학습에 필요한 문장을 biomedical abstracts로부터 추출하여

그림1. parsing tree 예

내용에 따라 '단백질-단백질 작용' 정보를 담고 있는 문장은 양(positive), 그렇지 않은 문장은 음(negative)의 label을 붙여 *supervised learning* 을 위한 학습 데이터를 구성한다.

다음으로 자연언어처리 분야에서 많이 사용되는 대표적인 태거(tagger)와 파서(parser)인 Brill's tagger 및 Collins Parser를 적용, 학습 예제에 대한 파싱 트리를 얻는다.

트리 커널 계산을 위해 앞서 구성한, 파싱 트리로 표현된 학습 예제를 적용하게 되는데 이때 순수한 문장을 파싱 트리로서 적용할 수도 있고, 이와 같은 순수한 문장의 파싱 트리 내에 존재하는 '단백질-단백질 작용' 관련 부분을 별도로 추출하여 서브트리로서 구성하고 이를 이용할 수도 있다. 또한 위에서 언급한 태거 및 파서 프로그램에서 제공하는 문법적인 POS(part-of-speech) 태거 대신 단백질에 대한 태그를 "PTN"으로 바꾸거나, 작용 관련 단어의 태그에 "-I"라는 표식을 첨가하여 POS 태그와 구분하는 과정을 적용할 수 도 있다. 이러한 추가 과정은 이를 적용하지 않은 과정의 결과와 비교를 위해 사용되었다.

마지막으로 계산된 트리 커널 Matrix값을 적용하여 대표적인 선형 분류기 학습 머신(linear classifier learning machine)인 SVM(Support Vector Machine [3])에 의해 학습과 함께 분류기로서의 성능을 평가하게 된다. 본 논문에서는 SVM을 구현한 대표적인 프로그램인 LibSVM을 이용, 학습과 성능 평가를 하였다. [2]

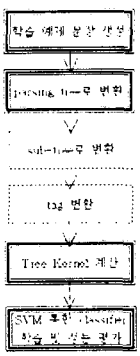


그림 2. 전체 process

4. 실 험

4.1 데이터

실험에 필요한 데이터 수집을 위해 생물학분야의 대표적인 온라인 서비스인 PubMed를 사용하였다. PubMed로부터 '단백질/유전자 작용' 관련 내용을 담고 있는 논문들의 abstract를 추출, 우선 단백질/유전자를 최소 두 개 이상 포함하고 있는 문장만 따로 분류하였다. 이들 가운데 세 개 이상 단백질/유전자 내용을 담고 있는 문장은 배제하거나 상호작용과 관련이 없는 일부 단백질/유전자는 'protein'으로 명칭을 변경하는 등의 작업을 거쳐 두 개만 포함하고 있는 문장을 학습 예제로서 선택하였다. 이는 이번 논문에서는 문장 내에 존재하는 두 개의 단백질/유전자에 대해 상호작용 정보를 담고 있는지 여부를 판명하는 것이 목적이기 때문이다. 그러나 이러한 방법은 실제 여러 단백질/유전자 쌍을 포함하고 있는 경우에도 문장 내에 존재하는 모든 가능한 쌍에 대해 적용함으로써 해당 문장 내에 상호작용 관련 정보 포함 여부를 판단하는데 유용할 것이다. 이런 과정을 거쳐 총 473개의 positive 예제와 259개의

negative 예제를 생성하였다.

4.2. 실험 결과

4.2.1 길이 및 빈도수가 유사한 문장 이용 실험 결과

positive 예제 군과 negative 예제 군을 모두 사용할 경우 각 경우마다 문장 개수 차이가 크고, 또한 문장 길이의 분포 특성이 다를 것으로 보고 먼저 이들 가운데 일부 데이터를 선별하여 학습에 사용하였다. 분포 특성을 조사하여 문장 길이와 문장의 개수가 반대 예제 군 에서도 비슷하게 대응하는 문장들을 중심으로 데이터를 선별한 결과 각각 98개의 문장으로 구성된 positive 예제군 및 negative 예제 군을 새롭게 구성하여 실험에 사용하였다.
태그변환에서 단백질에 대한 태그를 태깅과 파싱 결과로 불여진 POS 태그, "NN" 대신 "PTN"이라고 변형한 경우는 N, 그렇지 않은 경우는 Y로 표기하였다. 분류 성능은 LibSVM에서 학습하고 10-fold validation을 거쳐 얻은 결과를 각 5번씩 반복 수행한 결과의 평균으로서 다음과 같다.

Tag 변환	λ	(+)98:(-)98 sub-tree(+)			(+)98:(-)98 full-tree		
		acc. (%)	prec. (%)	rec. (%)	acc. (%)	prec. (%)	rec. (%)
N	0.2	82.12	83.81	85.99	77.55	86.96	79.59
	0.5	84.59	80.53	90.57	83.05	83.3	78.68
	1	71.63	75.31	86.53	67.04	62.04	100
Y	0.2	78.06	78.64	79.39	78.57	82.28	70.78
	0.5	84.59	85.43	79.32	82.35	85.12	79.23
	1	74.08	72.19	82.03	63.57	61.74	86.95

표 1. 길이 및 빈도수가 유사한 문장 이용 실험 결과

문장 전문을 그대로 트리로 구현한 것보다는 서브트리를 구성한 것이 전반적으로 계산 복잡도 뿐 만 아니라 보다 더 관심 정보를 구체적으로 담고 있다는 점에서 성능이 높을 것으로 예상했고 accuracy 측면에서 전반적으로 그러한 경향을 보였다. 특히 이러한 경향은 트리의 구조를 전부 반영하여 비교하는 경우인 $\lambda=1$인 경우에 대해 보다 명확하게 나타났다.
예상과 달리 단백질 태그를 "PTN"으로 변형한 결과가 그렇지 않은 경우에 비해 그다지 높지는 않았다. 이는 '단백질-단백질 작용' 내용을 담고 있는 문장 구조와 비슷하면서도 단백질에 대해 다른 화학물질이나 약품 등이 작용하는 내용을 담고 있는 그러한 negative 예제가 많이 있을 경우 효과적일 것으로 예상했으나 negative 예제 군에 그러한 예제가 많이 있지 않았거나 sub-tree를 구현하는 과정에서 그러한 구조가 선택되지 않았을 가능성도 배제할 수 없다.
주목할 점은 λ값의 변화에 따른 성능 변화인데, 전반적으로 범위 내의 너무 작거나 큰 값보다는 중간 정도 값에서 가장 높은 성능을 보이는 경향이 발견되었다. 이론적으로 λ값은 사이즈가 큰 트리를 down-weight 하기 위해 사용되는데 문제 및 예제 군의 특성에 따라 적절한 값의 선택이 있을 것으로 판단되며 중간 정도 값에서 가장 좋은 결과가 나온 점은 좀 더 연구가 필요할 것으로 보인다.

4.2.2 동일한 크기의 예제 군 및 전체 예제 군에 대한 학습결과

positive 예제 군에 비해 상대적으로 적은 수의 negative 예제 군 크기에 맞춰 positive 예제 군 내에서 임으로 259 개의 positive 예제를 추출하여 negative 예제 군과 함께 실험하였다. 또한 예제 군의 크기에 상관없이 생성된 예제 모두를 사용한 실험도 진행하였다. 문장 전문을 사용한 full-tree가 아닌 sub-tree에 대하여 "PTN" 태그 변환을 적용하여 앞서와 마찬가지로 10-fold cross validation하여 얻은 결과는 다음과 같다.

Tag 변환	λ	(+)259:(-)259			(+)473:(-)259		
		acc. (%)	prec. (%)	rec. (%)	acc. (%)	prec. (%)	rec. (%)
Y	0.2	83.78	85.03	82.28	85.66	87.58	90.91
	0.5	84.17	84.57	82.43	88.25	89.27	92.76
	1	75.48	81.81	64.17	74.32	71.85	99.14

표 2. 동일한 크기의 예제 군 및 전체 예제 군에 대한 학습결과

두 경우 모두, 98개의 예제로 구성된 예제 군을 사용한 이전의 결과에 비해 accuracy 면에서 전반적으로 성능 향상이 발견되었다. 예제 군 분석 결과 두 경우 모두 positive 예제 군에 비해 negative 예제 군에서 길이가 긴 예제가 좀 더 많이 발견되는 등 전반적으로 예제 군 특성의 차이가 있었던데 영향을 받았다고 본다. 또한 예상했던 바와 같이 positive 예제가 negative 예제에 비해 두 배 가까이 더 많이 사용된 두 번째 경우 positive 예제에 보다 편향된 학습으로 인해 recall rate가 전반적으로 90% 이상의 높은 수치를 나타낸 것으로 보인다.

5. 결 론

본 논문에서는 convolution kernel의 하나인 트리 커널을 이용한 커널 기반의 '단백질-단백질 작용' 내용 포함 문장의 추출 방법을 제시하였다. 최근 패턴 매칭 방법에서 좋은 성능을 보인 연구 결과를 참고하여 문장 구조적인 면에서 경향성 존재할 것을 예상, 파싱 트리 구조로서 문장을 표현하였고 트리 커널과 SVM을 이용하여 만족할 만한 분류 성능을 얻어낼 수 있었다. 관련 있는 내용 영역의 선택여부와 태그 및 문장 길이에 따른 성능의 변화가 관측되었으며, 단백질 이름에 대한 태그뿐 만 아니라 작용과 관련된 단어의 태그에 특수한 표식을 다는 등 semantic한 정보를 추가하기 위해 태그를 변환한 효과나 길이가 긴 트리에 불이익을 주는 λ의 기능을 구체적으로 파악하기 위한 연구[1]가 추후에 필요할 것으로 보인다.

감사의 글
본 연구는 과학기술부 국가지정연구실(NRL)사업에 의해 지원되었음.

참고문헌

[1] Alessandro Moschitti, *A study on Convolution Kernel for Shallow Semantic Parsing. In Proc. of the 42-th ACL-2004*, Barcelona, Spain, 2004
[2] Chin-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for SVM*, URL : http://www.csie.ntu.edu.tw/~cjlin/libsvm
[3] Cristainini, N. and Shawe-Taylor, J. 2000, *An introduction to Support Vector Machines and other Kernel-Based Learning Methods*, Cambridge University Press, 2001
[4] Jean- Michel, 2004, tutorial: Kernel Methods in Natural Language Processing, ACL-2004
[5] M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. In *Proc. of Neural Information Processing Systems* (NIPS' 2001).
[6] Minlie Huang, Xiaoyan Zhu, Yu Hao, Donald G. Payan, Kunbin Qu and Ming Li. 2004, Discovering patterns to extract protein-protein interactions from full texts, Bioinformatics, vol.18, pp.155-161
[7] N. Cancedda, E. Gaussier, C. Goutte, and J.M. Renders. 2003. Word-Sequence Kernels. *Journal of Machine Learning Research*
[8] Park,J.C., Kim,H.S. and Km,J.J. 2001 Bidirectional incremental parsing for automatic pathway identification with combinatory categorical grammar. *In proc. of the Pacific Symposium Biocomputing*, Hawaii, USA, pp.396-40