Brain's Pick : 단어 간 유사도 파악 방법

오디오와 이미지는 숫자로 이뤄져 있다. 반면 텍스트는 문자다. 0, 1만을 인식하는 컴퓨터가 어떤 단어를 이해하도록 하려면 이 텍스트를 숫자로 변환해야 한다. 그러나 이 작업은 쉽지 않다. 그래서 이런 자연어를 컴퓨터가 잘 이해하도록 하고자 생겨난 연구 분야 중 하나가 바로 자연어 처리(natural language processing, NLP)다. 그렇다면 어떻게 이 텍스트 데이터를 유의미한 숫자로 바꿔볼 수 있을까?

글 | 이수경 samantha.lee@kakaobrain.com

2016년 3월 알파고와 이세돌 9단이 펼치는 세기의 대결을 두 눈으로 목도한 이후 인공지능을 제대로 공부해 봐야겠다고 결심했습니다. 인공지능 연구의 본진인 카카오브레인으로 걸어 들어온 이유입니다. 인공지능 기술과 이로 인해 바뀔 미래 사회를 다루는 글을 통해 사람들과 소통하고 싶습니다.

감수 | 이주진 zeze.zzz@kakaobrain.com

사람들의 삶을 바꾸는 소프트웨어를 만들고 싶었던 대학생이 자연어 프로젝트를 하는 과정에서 인공지능을 만났습니다. 현재 카카오브레인에서 인공지능 리서치 엔지니어로 활동하고 있습니다. 사람을 이해하는 인공지능, 모두에게 더 나은 교육 경험을 제공하는 인공지능을 만들기 위해 노력하고 있습니다.

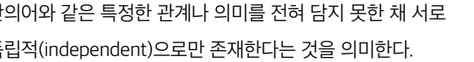
감수 | 임성빈 leo.brain@kakaobrain.com

수식과 증명을 사랑하던 수학자가 로보틱스와 인공지능의 매력에 빠지게 되었습니다. 카카오브레인에서 수학과 엔지니어 사이의 가교 역할을 담당하는 리서치 사이언티스트로 활동하고 있습니다. 안전하고 신뢰도 높은 인공지능을 개발해 우리 사회를 보다 윤택하게 만들고 싶습니다.

원-핫 인코딩(one-hot encoding)

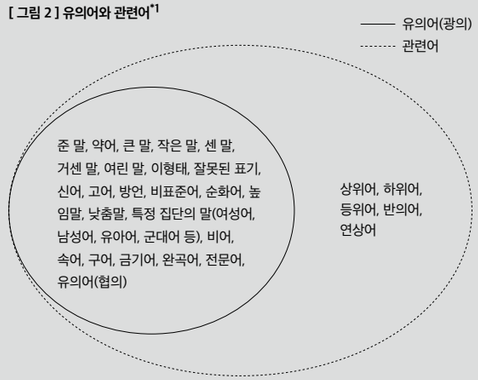
텍스트를 유의미한 숫자(벡터)로 바꾸는 가장 손쉬운 방법론은 바로 '원-핫 인코딩(one-hot encoding)'이다. 이는 N개의 단어를 각각 N차원의 벡터로 표현하는 방식이다. 단어가 포함되는 자리엔 1을 넣고 나머지에는 0을 넣는다. 사전이 [인간, 펭귄, 문어, 사람]이라면 인간을 표현하는 벡터는 [1, 0, 0, 0]이 되는 식이다. 단어 하나에 인덱스 정수를 할당한다는 점에서 '단어 주머니(bag of words, BoW)'라 부르기도 한다.

[그림 1] 원-핫 인코딩은 한 개의 요소만 1이고 나머지는 0인 N차원의 벡터로 표현된다



이 방식은 나름대로 좋은 성능을 내고, 지금까지도 사용하는 사람들이 많지만 아주 큰 단점이 있다. 바로 컴퓨터가 단어의 의미 또는 개념 차이를 전혀 담지 못한다는 것이다. 예를 들어, '과학'과 '공학'의 관계는 '과학'과 '수박'의 관계와 차이가 없다.

[그림 2] 유의어와 관련어[*1]



수학적으로 보자면, 원-핫 벡터들은 딱 하나의 요소만 1이고 나머지는 모두 0인 희소 벡터(sparse vector) 형태를 띤다. 이런 경우 두 단어 벡터의 내적(inner product)은 0으로 직교(orthogonal)를 이룬다. 이는 단어 간 존재하는 유의어, 반의어와 같은 특정한 관계나 의미를 전혀 담지 못한 채 서로 독립적(independent)으로만 존재한다는 것을 의미한다.

[그림 3] 원-핫 벡터 방식으로는 '과학'과 '공학' 간의 개념 유사성을 발견할 수 없다.

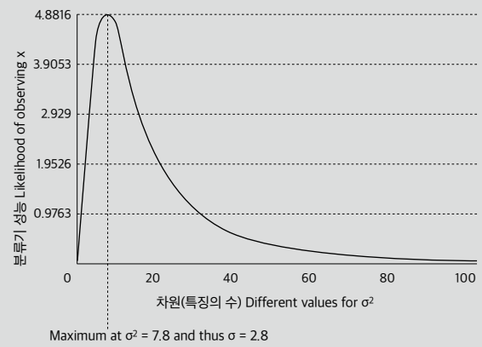
'차원의 저주(curse of dimensionality)' 문제도 발생한다. 하나의 단어를 표현하기 위해 말뭉치(corpus)에 존재하는 수만큼의 차원을 가지게 되면 계산 복잡성이 기하급수적으로 늘어난다. 예를 들어, 40만 개의 고유의 언어 데이터 셋을 활용해 원-핫 인코딩 배열을 구성한다면 그 차원 수는 40만에 이르게 된다.

[표 1] 단어 수가 많아질수록 차원의 크기는 기하급수적으로 증가한다.[*2]

단어 활용 상황	차원(단위: 만)
음성	2
PTB	5
대사전	50
구글 웹크롤 말뭉치(1TB)	1300

하지만 차원 수가 일정 수준을 넘어서면 분류기(classifier)의 성능은 되려 0으로 수렴한다는 점에서 봤을 때 제아무리 뛰어난 성능을 가진 컴퓨터라도 이런 고차원(high dimensionality)의 벡터를 학습하기는 어렵고 성능이 떨어지기 마련이다.

[그림 4] 특징 수가 일정 수준을 넘어서면 분류 성능은 오히려 낮아진다



단어 임베딩(word embedding)

이에 학자들은 의미를 최대한 담아 단어를 벡터로 바꾸는 단어 임베딩 모델[*3]을 고안하게 된다. 이를 위해 오늘날 통계적 자연어 처리에서 가장 성공한 아이디어로 손꼽히는 분산 표상(distributed similarity based representation)[*4]이라는 개념이 차용됐다. 어떤 사람의 주위 사람을 보면 그 사람의 됨됨이나 가치관을 미루어 짐작할 수 있듯이 비슷한 분포를 가진 단어의 주변 단어들도 비슷한 의미를 가진다는 것을 말한다. 예를 들어, '주어-[ ][ ]-공부했다'라는 문장 구조에서 [ ][ ]에 '과학'과 '공학'이 모두 들어갈 수 있다면 과학과 공학이 유사한 역할을 한다고 유추할 수 있다.

원-핫 인코딩과는 달리, 분산 표상에서는 하나의 단어가 미리 정의된 차원(보통 20~200)에서 연속형의 값을 갖는 벡터로 표현된다. 몇 천, 몇 만 차원이 필요했던 희소 벡터보다 훨씬 적다.

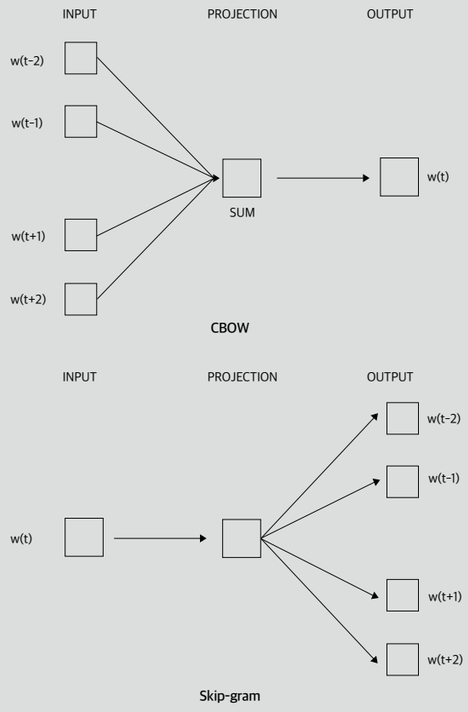
게다가 각각의 차원은 모두 정보를 들고 있다. 벡터 연산을 통해 단어 벡터 간 유사도를 구할 수 있다는 의미다. 즉, 단어 임베딩은 '사람' 벡터가 '인간' 벡터와 얼마나 유사한지, '문어' 벡터와는 얼마나 다른지 벡터 연산으로 단어 간 관계를 추론할 수 있다는 의미다.

서로 비슷한 벡터로 표현되는 단어에 대해선 일반화된 학습도 가능하다. 즉, '인간'이라는 단어를 학습했다면 '사람'이라는 단어에도 비슷한 학습 방식을 적용할 수 있다.

학습 데이터 양이 작으면 이런 관계를 추론하기가 힘들 수는 있으나 수많은 문서를 학습하다 보면 이런 단어 간 관계를 보다 정확하게 파악할 수 있다.

이처럼 단어를 벡터로 임베딩하는 방식은 머신러닝을 통해 학습된다. 2000년대에 'NNLM(neural network based language model)' 방법론이 고안됐다. 이는 요수아 벤지오(Yoshua Bengio) 교수가 제안한 모델로, 신경망에 기반해 단어를 벡터로 바꾸는 방법론으로 주목받았다. RNNLM(recurrent neural network language modeling)은 NNLM을 업데이트한 버전이다. 이것이 CBOW(continuous bag of words)[*5]와 Skip-gram(SG)[*6]이라는 아키텍처로 다시 한번 진화해 현재의 Word2Vec로 이어졌다. 그 외 GloVe, FastText과 같은 방법론도 있다.

[그림 5] Word2Vec 구조[*7]



카카오브레인의 WordRep

국내에는 다양한 목적의 어휘 관계 사전이 개발되고 있다.[*8] 한국어 교육 목적으로 개발된 '학습자를 위한 한국어 유의어 사전'과 (주)낱말의 낱말망, 유의어 사전, 반의어 사전, 스마트 시소러스(Smart Thesaurus) 등이 그것이다. 민간 기업인 (주)낱말에서는 6,625쪽에 달하는 분량의 '넓은 풀이 우리말 유의어 대사전'[*9]을 집필하기도 했다.

다만 이런 대규모 사전을 편집하는 데 투입되는 인력과 자금이 많지 않아 시간이 오래 걸린다는 문제가 있다. 아울러 인간 편집자의 주관에 의해 단어 간 관계가 설정될 가능성도 완전히 배제할 수 없다. 조금 더 실효성 높은 단어 간 관계 데이터를 제공하자는 취지에서 카카오브레인은 관련 프로젝트를 진행했다. 유의어와 반의어에 대해 알고 싶은 전 세계 작가나 신문 기자 등을 위해 다국어 서비스를 만들겠다는 것이 카카오브레인의 프로젝트가 이루고자 하는 목표였다. 그렇다면 카카오브레인에서는 어떤 과정을 거쳐 단어 간 유사도를 표현해냈을까?

1) 단어 벡터 사전 수집

말뭉치의 출처는 해결하려는 문제에서 사용하는 자연어 데이터에 따라 달라진다. 비교적 수월하게 대량의 텍스트 데이터를 수집할 수 있다는 이유로 통상적으로 위키피디아(Wikipedia) 말뭉치를 활용한다. 하지만 매 번 자연어 문제를 풀 때마다 단어 임베딩을 수행하는 것은 매우 비효율적인 작업이다. 이러한 이슈를 해결하고자 페이스북은 자체 단어 임베딩 기법인 FastText[*10]를 이용해 만든 294개 언어의 벡터 사전을 자사 깃허브 저장소(GitHub Repository)[*11]에 공개했다. 카카오브레인의 WordRep 프로젝트에서는 앞서 공개된 벡터 사전을 이용해 단어의 유사성을 보여준다.

2) 자원 절약

공개된 294개 언어의 벡터 사전을 바탕으로 단어들 사이의 관련성을 검색하려면 서비스를 운영할 서버가 294개의 사전 데이터를 전부 보유하고 있어야 한다. 문제는 파일 크기였다. 영어는 6.2GB, 한글을 포함한 주요 언어들은 1~3GB, 그 외 언어들은 수십~수백 메가바이트(MB)의 용량을 차지했다. 현실적으로 이 큰 용량의 파일들을 전부 디스크와 메모리에 올려서 서비스하는 것은 불가능하다. 보통 서비스에서 인스턴스(instance)에 할당하는 메모리 크기가 512MB~2GB면 충분하다는 점을 봤을 때 100GB가 넘는 메모리 크기는 상당한 자원 낭비다.

카카오브레인 연구팀에서는 자연어 처리에 사용되는 파이썬(Python) gensim 라이브러리를 이용해 단어 사전을 로드할 때 각 언어에서 가장 많이 사용되는 단어 일부를 추출했다.[*12] 이렇게 로드한 gensim 모델을 다시 pkl로 저장해 개별 사전의 용량을 줄였다. 그 결과, 영어 모델은 6.2GB에서 600MB까지 줄어들었고, 한글을 포함한 9개 언어는 1~3GB에서 240MB로, 나머지 언어들도 최대 60MB까지 축소됐다.

비단 용량 감축뿐만 아니라 사전 로드 시간도 획기적으로 단축됐다. 기존 vec 파일을 활용해 하나의 언어 사전을 로드할 때 1분 가까이 걸리던 것이 pkl로 바뀌면서 2~3초 만에 로드됐다. 이렇게 로드 속도가 수 초 단위로 줄어들면서 모든 언어를 미리 로드하지 않아도 됐다. 10개 언어를 제외한 언어에 대한 검색을 요청할 때마다 서버로부터 데이터를 로드하면서 메모리 이슈도 자연스럽게 해결했다.

3) 검색 및 시각화

사용자가 언어와 단어를 입력한 뒤 유사 단어를 찾도록 요구하면 다음과 같은 과정을 거쳐 시각화(visualization)가 이루어진다. 2~4번 과정이 어떻게 이루어지는지 더 자세하게 소개하고자 한다.

유사 단어 검색 및 시각화 과정

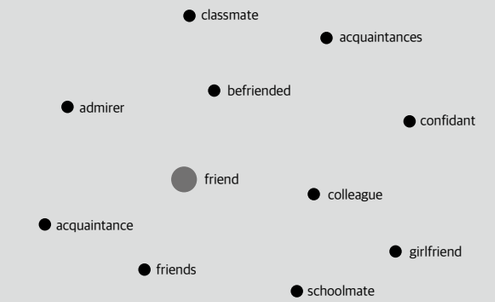
(1) 선택한 언어의 사전이 로드되어 있는지 확인 후, 로드되지 않은 경우 로드한다.
(2) 원하는 수의 유사 단어를 찾는다.
(3) 검색한 단어와 유사 단어들의 벡터들을 2차원으로 차원 축소한다.
(4) 2차원 좌표계를 정규화(normalization)한 뒤 웹(web)에 그려낸다.

우선 유사한 단어를 찾는 과정은 앞에서 사용한 gensim 라이브러리에서 'most_similar' 함수로 쉽게 찾을 수 있다. 이 함수를 이용하면 단어와 유사도[*13]들의 배열이 반환되는데, 여기서 단어들만 가져와 사전에서 해당하는 벡터를 가져온다.

이후 scikit-learn에 있는 tsne[*14]를 이용하여 단어 벡터를 2차원으로 축소한다. tsne는 차원 축소 알고리즘으로 고차원의 벡터들 사이의 상관 관계를 저차원에서 최대한 유지해준다. 이런 방식으로 차원을 2차원으로 축소하면 고차원에서 나타났던 단어들 사이의 거리가 거의 손실되지 않고 반영된다.

이렇게 원본 단어와 유사 단어들의 2차원 좌표 값을 얻어내면 웹에 픽셀 단위로 그려낼 수 있도록 두 개의 축에 해당하는 값들을 50~250 사이로 설정한다. 단어와 좌표 값으로 구성된 배열들은 파이썬 플라스크(Python Flask)[*15]의 템플릿(template)으로 전달되고, HTML의 SVG 태그 안에서 각각 하나의 점으로 그려진다.

[그림 6] WordWeb에서 'friend'를 검색한 결과[*16]



4) 한계 및 개선 방향

현재 WordWeb 서비스는 뚜렷한 한계를 가지고 있다. 가장 쉽게 확인할 수 있는 문제는 한글 단어를 입력하게 될 경우 유사한 단어가 아닌 다른 형태소와 조합된 결과물들이 나타난다는 것이다. 예를 들어, '사랑'을 검색하면 '사랑을', '사랑과', '사랑에서'와 같이 단순히 조사를 붙인 어절들이 결과로 나타난다.

[그림 7] WordWeb에서 '사랑'을 검색한 결과[*17]

통상적으로 한국어 자연어 처리 과정에서는 형태소 분리를 통해 문제를 해결하는데, 페이스북이 제공하는 벡터에서는 관련 처리를 수행하지 않은 것으로 보인다. 이는 한국어뿐만 아니라 일본어, 중국어 등에서도 동일하게 나타나는 현상이다. 카카오브레인 연구팀은 향후 관련 처리를 모두 반영한 벡터 사전으로 교체하거나

후처리를 통해 동일한 단어들을 제외하는 등의 방법으로 보다 나은 결과물을 내는 데 집중할 계획이다.

임성빈 연구원님이 힐베르트 공간에 대한 수학적 내용을 설명했다. 이 글을 읽고 다시 앞으로 돌아가면 단어 임베딩과 머신러닝에 대한 내용을 조금 더 폭넓게 이해할 수 있을 것이다.

힐베르트 공간(Hilbert Space)

임성빈 카카오브레인 연구원

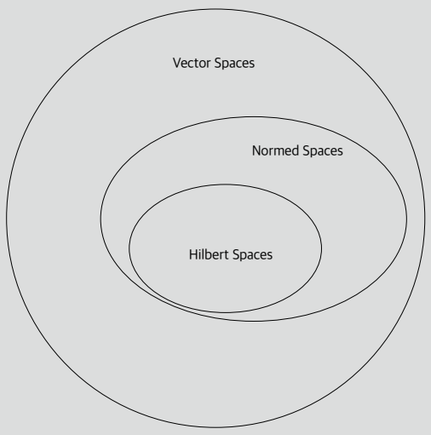
머신러닝 · 통계학을 공부하는 많은 사람들은 벡터 공간(vector space)을 기하적인(geometric) 의미로 인식한다. 하지만 굳이 자세히 보자면 벡터는 덧셈(addition)과 스칼라 곱(scalar multiplication)에 관한 몇 가지 규칙으로 정의된 대수적(algebraic)인 공간에 더 가깝다. 이는 선형대수 첫 번째 강의 시간에 배우는 내용이기도 하다. 좌표 공간뿐만 아니라 함수, 확률변수, 행렬 등도 벡터 공간으로 표현해볼 수 있다.

이처럼 다양한 개념을 벡터 공간으로 추상화하는 이유는 좌표 공간에서 주로 사용하던 선형대수 기법들을 손쉽게 적용할 수 있어서다. 양자역학에서 자주 쓰이는 브라-켓(bra-ket) 표기법이 대표적인 예다. 이 표기법은 양자 상태(quantum state)를 기술하는 파동함수(wave function)의 모임을 벡터 공간으로 보고 선형대수 연산 규칙을 그대로 따른다. 신호 처리 분야에서는 센서를 통해 관측되는 신호[*18]를 벡터로 처리하고 행렬로 정의된 여러 연산자(operator)를 적용하기도 한다.

벡터 공간을 기하적인 의미로 받아 들이려면 '내적'이 필요하다. 이는 우리에게 익숙한 '각도(angle)'를 정의할 수 있다는 점에서 사영(projection)과 관계가 있다. 한편, 확률변수의 공간에서 이 각도는 상관계수(correlation)를 의미한다. 정확히 말하면 각도의 코사인값이 상관계수와 동일한데, 확률변수를 벡터로 받아 들인다면 두 개념 사이의 관계를 직관적으로 상상해볼 수 있다. 예를 들어 두 확률변수의 상관계수가 1이면 각도상으로는 0도, 즉 같은 방향을 가리킨다는 의미다. 만약 두 확률변수가 서로 독립(independent)이면 상관계수는 0으로, 각도상으로는 서로 직교인 관계가 된다.

아울러 벡터 공간은 내적을 이용해 거리 위상(metric topology)까지 갖출 수 있다. 바로 내적의 절대값을 활용해서 구한 거리보다 더 강한 개념의 '노름(norm)'을 통해서다. 여기서부턴 해석학에서 다루는 극한(limit)을 이야기할 수 있다. 극한은 우리가 잘 아는 미적분이나 SGD(확률적 경사하강, stochastic gradient descent) 처럼 해(solution)를 찾는 기법에서 꼭 필요한 개념이다.

[그림 8] 벡터 공간과 노름 공간, 힐베르트 공간 간 관계[*19]



거리 위상을 갖춘 벡터 공간에서 극한을 제약 없이 활용하려면 완비성(completeness)이란 성질을 갖춰야 한다. 완비성은 수학적으로는 다소 복잡한 개념[*20]인데, 간단히 설명하자면 공간에 빈 틈(gap)이 없어야 함을 뜻한다. 극한을 활용하는 대부분의 수학/과학/공학 이론은 완비성을 전제하고 있다.

완비성을 갖춘 내적이 정의된 벡터 공간은 통계학/머신러닝의 가장 기본적인 시작점이라 할 수 있다.[*21] 수학자들은 이 공간을 힐베르트 공간(Hilbert space)[*22]이라 부른다. 요컨대 힐베르트 공간은 우리에게 친숙한 유클리드 공간을 일반화한 것이다.[*23] 이쪽 분야를 연구하는 수학 이론을 함수해석학(functional analysis)이라 부른다.

그렇다면 내적, 각도, 거리, 극한의 개념을 모두 갖춘 힐베르트 공간과 유클리드 공간의 차이는 무엇일까? 신호 처리에서 푸리에(Fourier)나 웨이블릿(wavelet) 해석을 다뤄봤거나 물리학에서 열 방정식(heat equation) 혹은 파동방정식(wave equation)을 공부한 사람이라면 각 변환이나 방정식의 해를 표현하는 기저 함수(basis function)의 개수가 무한대라는 사실을 기억할 것이다. 즉, 힐베르트 공간은 무한 차원(infinite dimension)을 허용한다. 선형대수학이 유클리드 공간 같은 유한차원 공간을 주로 다뤘다면 함수해석학은 무한 차원을 가진 공간상의 선형대수학인 셈이다.

1 논문 | 한유석. (2014). 한국어 유의어사전 구축에 관한 연구. 언어학, 22(1), 169-182. **2** 참고 | http://stanford.edu/class/cs224n/lectures/lecture3.pdf **3** 참고 | 또는 희소(sparse)의 반댓말인 밀집(dense)이라는 표현을 활용하기도 한다. 원-핫 인코딩처럼 대부분이 0인 벡터가 아닌, 모든 차원이 값을 가지고 있어서다. **4** 참고 | 영국 언어학자인 존 퍼스(John Rupert Firth, 1890-1960)는 단어의 주변을 보면 그 단어를 안다(You shall know a word by the company it keeps)고 말했다(1957). **5** 참고 | 타깃 단어를 이용해 주변 단어를 예측하는 방식 **6** 참고 | 주변 단어를 이용해 타깃 단어를 예측하는 방식 **7** 논문 | 임미선, 강신재. (2017). 단어 쓰임새 정보와 신경망을 활용한 한국어 Hedge 인식, <예술인문사회융합멀티미디어논문지>, 7권 9호, 317-325. **8** 참고 | 안의정. (2015). 사전과 정보란 무엇인가, 커뮤니케이션북스. **9** 참고 | 표제어가 10만 단어, 1차 유의어가 28만개, 2차 유의어가 200만개에 이른다. 반대말, 방언, 외래말, 높임말도 함께 수록돼 있다. **10** 논문 | Bojanowski, P., Grave, E., Joulian, A., & Mikolov, A. (2016). Enriching Word Vectors with Subword Information. arXiv:1607.04606 **11** 참고 | https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md **12** 참고 | 영어는 50만 단어, 한글을 포함한 9개 언어는 20만 단어, 그 외의 언어는 5만 단어로 제한했다. **13** 참고 | 0~1 사이의 실수가 반환된다. **14** 참고 | Van der Maaten, L. (2015). Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research. 15. 3221-3245. **15** 참고 | 파이썬을 이용해 웹을 구축하게 하는 프레임워크다. http://flask.pocoo.org **16** 참고 | http://wordrep.zeze.dev.9rum.cc/wordweb/en/friend/10 **17** 참고 | http://brain-cluster-gpu14.dakao.io:1041/wordweb/ko/사랑/20 **18** 참고 | 이 신호들은 노이즈(noise)가 끼기 마련이므로 확률변수가 된다. **19** 참고 | https://archive.cnx.org/contents/6d15563e-d611-4d75-b5a7-34c937a1b174@8/common-hilbert-spaces **20** 참고 | 수열 간의 거리가 0으로 수렴하는 코시(Cauchy) 수열은 반드시 공간의 어느 한 점으로 수렴한다는 의미를 갖는다. **21** 참고 | 위키백과에서 통계적 학습(statistical learning)을 검색해보면 통계학과 함수해석학의 응용이라고 검색된다. **22** 참고 | 현대 수학의 아버지 다비트 힐베르트(David Hilbert, 1862 - 1943)의 이름을 기리기 위해 붙여진 이름이다. **23** 참고 | 이는 천재 수학자 폰 노이만(John von Neumann, 1903 - 1957)의 업적이다.