

카카오미니는 어떻게 스무고개 정답을 맞출까

올해 봄, 가수 김정치마가 낸 3집 앨범 TEAM BABY의 타이틀 곡 '나랑 아니면'은 "야 나랑 놀자 밤늦게까지 함께 손뼉 치면서" 라는 가사로 시작한다. 호모 루덴스(Homo Ludens). 이처럼 우리는 누구나 즐겁게 놀고 싶은 욕구를 가지고 있다. 어린 시절부터 지금까지 놀이기구를 타거나 블록 쌓기, 퍼즐 맞추기, PC게임, 모바일게임 등의 '놀이'를 끊임없이 해 오고 있는 것이다. 때문에 그러한 '놀이'가 새로운 시장을 만들어 나가고 있는 인공지능 스피커에 들어가는 것은 어찌보면 당연한 일이다.

AI스피커에 왜 게임이 필요할까?

우리가 꾸준히 놀이를 하는 이유를 생각해보면, 그를 통해서 '재미'를 얻기 때문이다. 카카오미니가 담을 수 있는 수많은 기능 중에서 게임은 놀이를 통한 재미 제공에 목적을 두고 있고, 교육용 게임에선 학습도 덩으로 얻을 수 있다.

우리는 유아기 때부터 이런저런 스킬을 얻고 있는데, 그 중 카톡 보내기보다 먼저 배우는 것이 말하기다'. '엄마', '아빠', '맘마'로 시작해서 발전해 나가지 않나. 외국어를 배울 때를 생각해보자. 가장 먼저 트이는 것은 듣기, 그 다음은 말하기, 그 다음이 바로 쓰기의 순서다.

'카카오'라는 브랜드는 스마트폰과 함께 카카오톡을 사용해온 대중들에게는 가깝고 친근한 이미지로 자리잡았을 것이다. 카톡 사용 집단의 범위가 넓다는 것을 고려해볼 때, 카톡보다 나중에 나왔지만 음성으로만 이루어지는 카카오미니를 완벽한 문장을 구사하는데 어려움이 있는 아이들부터 휴대전화를 잘 다룰 줄 모르는 분들도 사용할 수 있었으면 좋겠다. 예를 들면 카톡보단 아직 전화가 편한 우리 어머니와 같은 분들.

"나랑 놀자!"를 하기까지

정보만 전달하는 스피커도 좋지만 대화를 주고받고 같이 놀 수 있어야 한다고 생각한다. 우리들은 다른 사람들과의 관계에서 정보전달도 많이 하지만 함께 공감을 나누거나 놀면서 보다 가까워지기 때문이다.

예를 들어, 학교에 새롭게 입학한 에이미라는 친구가 있다. 기존에 알던 친구라곤 한명도 없는데, 아뿔싸 딱 생각을 하다가 선생님이 숙제 내주시는 걸 못 듣고 말았다.

정보전달 - 용기를 내어 옆자리의 라이언에게 "오늘 숙제가 뭐야?" 라고 물어보고, 라이언은 에이미에게 숙제가 뭐였는지 알려준다.
소통 - 그 다음 날, 에이미는 라이언과 인사를 하고 어제 숙제가 어려웠다는 내용의 대화를 한다.
재미 - 이후, 에이미와 라이언은 가까운 친구가 되어 같이 술래잡기도하고 얼음땡도 하는 사이가 되었다.

이렇게 가까워지는 단계를 보면 순서가 정보전달 → 소통 → 재미 의 순서가 있는데, AI스피커가 친구가 되려면 역시나 이런 순서를 따르는 것이 좋지 않을까? 그렇다면 카카오미니에 게임을 넣어보자.

WHY 스무고개?

그 많은 게임 중에 왜 스무고개를 넣었는가에 대해 묻는다면, 그 답은 첫 스타트를 끊기 '쉬워 보여서'였다. 답변의 종류가

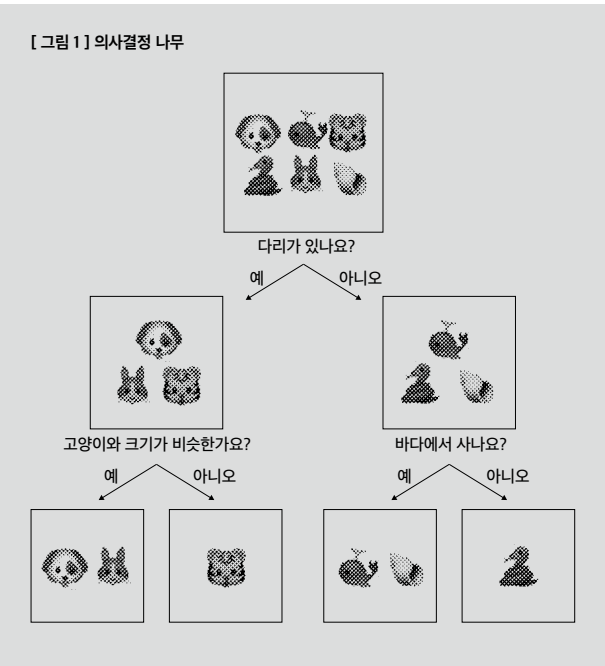
'예/아니오(YES/NO)' 두 가지로만 이루어져 있어 음성 입력이 쉽고*, 그에 따라 게임을 진행하다 보면 답이 쉽게 나올 것이라고 생각했기 때문이다.

어떤 방법으로 문제를 맞출 수 있도록 해야 하나

스무고개 게임에서 알고리즘의 핵심은 '어떻게 하면 답을 잘 맞추게 할 수 있을것인가'이다. 더 정확히 말하면 '어떻게 하면 답을 더 빨리, 더 정확하게 맞추게 할 수 있을까'이다. 많이 쓰이는 방법을 생각해보면 가장 단순하게는 의사결정 나무(decision tree)부터 랜덤 포레스트(random forest), 나이브 베이즈(naive bayes), 뉴럴 네트워크(neural network) 등을 생각해 볼 수 있다.

의사결정 나무

처음엔 '예/아니오(YES/NO)'로만 답이 이루어지기 때문에 당연히 의사결정 나무(decision tree)면 될거라는 순진한 생각을 했다. 스무고개의 원리 자체는 이진검색(binary search)과 유사한 방식이다.



의사결정 나무에서도 가장 단순하게 생각해볼 수 있는 것은 고정된 트리가 딱 하나만 존재해서 질문의 순서가 정해져 있는 의사결정 나무이다. 예를 들어, 동물 스무고개를 제공하는 웹사이트인 애니멀게임³ 에서의 첫 번째 질문은 항상 "Can it fly?(날 수 있나요?)" 이다. 여기서 '예(YES)'를 누르면 "Does it have feathers?(깃털이 있나요?)", '아니오(NO)'를 누르면 "Is it a very

글 | 조희주 amy.wine@kakaocorp.com

통계학과 졸업 후 머신러닝의 맛을 보다가 개발에 흥미가 생겨 능력자들 사이에서 학교다닐 때 안하던 공부중인 개발꿈나무 입니다.(라고 쓰고 우주먼지라고 읽습니다) 머신러닝을 적용한 프로젝트를 제 맘대로 만들어 내는게 단기적 꿈이고 세계평화와 인류의 행복을 위한 인공지능을 만드는 것이 장기적 꿈입니다.

글 | 김수형 cantabile.58@kakaocorp.com

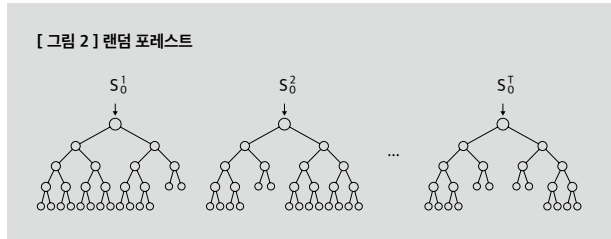
이창호를 능가하는 인공지능 바둑을 만들고 싶었으나 딥마인드에 선수를 빼앗겨 억울해하고 있는 늑은 개발자입니다. 한 때, 대학에서 학생들에게 수학과 프로그래밍을 가르쳐보기도 했지만 여전히 수학은 너무 낯설고 프로그래밍은 너무 어렵지만 합니다.

small animal?(덩치가 매우 작은가요?)"가 나온다. 세상 노잼… 이런 게임이라면 금방 싫증이 날 것이다.

질문 순서에 랜덤 요소를 반영하고 동적으로 트리를 새로 생성해 나간다면 이런 문제는 해결할 수 있을 것이다. 하지만 이상적인 경우엔 잘 동작해야만 하는 의사결정 나무 모델은 유저가 답변을 한번이라도 다르게 말하거나 인식이 잘못되면 돌아올 수 없는 강을 건너게 된다. 이후로는 어떤 질문을 아무리 많이 하더라도 답을 맞출 수 없다.

예를 들어, 위의 [그림1]에서 '호랑이'를 생각한 유저가 실수로 혹은 잘 모르고 "고양이와 크기가 비슷한가요?"라는 질문에 '아니오'가 아닌 '예'로 대답을 해버렸다면? 의사결정 나무에서 한번 뺏은 가지는 다시 뒤로 돌아오지 않기때문에 이제 유저의 답은 영영 나오지 않게 된다. 스무고개, 만만치 않다.

Random Forest⁴



그런 의사결정 나무의 단점을 극복하기 위해 의사결정 나무를 여러 개 만들어 숲(forest)을 이뤄서 경우의 수를 대폭 확장한다는 개념의 랜덤 포레스트(Random Forest)를 써볼 수 있다⁵.

하지만 랜덤 포레스트에서 데이터셋의 크기가 작은 경우엔 오버피팅(over-fitting)될 확률이 높다. 즉, 너무 구체적으로 디테일하게 학습돼 버리는 것이다. 또한 새로운 데이터가 반영될 때마다 숲(forest)을 다시 만들어 줘야(rebuild)해 풀고자 하는 문제에 비해 계산량이 너무 많아진다.

Neural Network

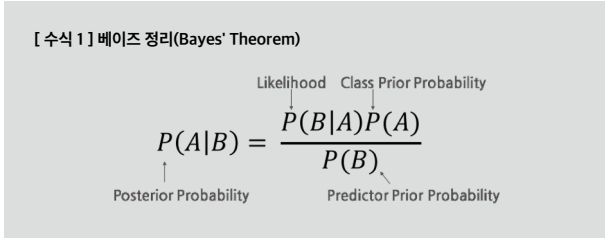
스무고개를 뉴럴 네트워크로 진행할 수도 있다. 로빈 부르게너(Robin Burgener)가 1988년 부터 뉴럴 네트워크를 이용해서 만든 스무고개는 웹⁶이나 애플리케이션에서 해볼 수 있고, 이 원리는 특허⁷로 등록되어 있다. 이 원리는 2006년 미국항공우주국(NASA) 주최의 콜로키움에서 발표⁸된 적도 있다. 아마도 현존하는 가장 크고 오래된 인공지능 스무고개일 것이다.

하지만 직접 들어가서 게임을 해보면 보이듯 선택지가 기존에 알던 '예/아니오(YES/NO)'가 아니고 문제에 따라 선택

대안이 바뀐다. 오랜 시간 많은 데이터를 쌓았겠지만 아쉽게도 우리가 원하는 품질과 방향이 아니었다. 작고 제한된 데이터셋(data set)이지만, 더 친근하고 재미있는 스무고개를 만들고 싶었기 때문이다.

베이즈 모델(Bayes' Model)

랜덤 포레스트의 오버피팅 문제를 걱정하지 않아도 되고 실시간으로 변하는 데이터를 적용시킬 수 있는 솔루션으로 나이브(naive; 꾸밈 없는, 자연스러운) 베이즈가 있다. 나이브 베이즈 분류(naive bayes classification)⁹라는 이름으로 머신러닝에서 많이 쓰이고 있는 방법으로, 상황에 따라 랜덤 포레스트보다 편리할 수 있다.



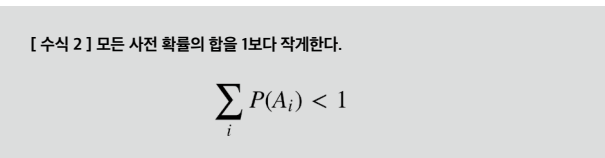
컨셉은 베이즈 정리를 꾸밈없이 담백하게 사용해서 확률을 계산하는 것이다. 베이즈 정리가 생소하다면 이것만 기억하면 된다¹⁰. "사후 확률(posterior probability)은 사전 확률(prior probability)과 우도(likelihood)의 곱이다." 즉, '일이 발생하기 전의 확률'과 '일이 발생할 가능성의 정도'를 곱하면 '일이 발생한 후의 확률' 인 것이다.

카카오미니가 스무고개의 답을 맞추는 과정

카카오미니의 스무고개는 베이즈 정리를 기반으로 한 알고리즘을 사용하고 있다. 앞서 언급했듯이 베이즈 정리는 우도와 사전확률의 곱이고, 더 간단히 말하면 전체의 확률에서 그 사건이 일어날 확률이다. 아래의 예시를 보면 좀 더 구체적으로 이해가 될 것이다.

사후분포를 구하기 위한 준비물에는 P(A), P(B|A), P(B)가 필요하다 차례대로 준비해보자

먼저 P(A)는 각 동물에 대한 사전 확률이다. 게임의 플레이어가 얼마의 확률로 그 동물을 생각할지를 나타낸다. 초기 사전 확률은 임의로 지정했으며, 이는 데이터가 충분히 쌓이면 그 분포에 기반해 변할 것이다. 이 사전 확률의 총 합은 1보다 작다. 1이 아니라 1보다 작은 이유는 우리가 게임 속에 넣어 둔 동물이 사용자가 정답으로 생각할 수 있는 세상의 모든 동물을 완벽하게 넣은 것이 아니기 때문이다. 식으로 나타내면 [수식 2]와 같다.



이 조건에서, 각 동물의 사전 확률을 지정해준다. 여기서 사전 확률의 총 합은 0.8(=80%)로 두자. 강아지나 토끼 등 비교적 친근한 동물은 선택될 확률이 높고, 소라나 뱀 등 친근하지 않은 동물은 선택될 확률이 낮다.

[표 1] 사전확률(Prior Probability)

Prior	강아지	토끼	고래	호랑이	소라	뱀
P(A) _(i = 강아지, 토끼, ..., 뱀)	0.2	0.2	0.1	0.1	0.1	0.1

이제 우도인 P(B|A)를 준비할 차례다. A인 경우에 B일 확률로, A는 'A동물을 정답으로 생각하고 있는 경우', B는 'B질문에 대해 '예'라고 할 경우'로 본다. 나이브 베이즈 모델을 적용하기 위한 필수 조건으로 각각의 확률은 독립적(independent)이라고 가정한다.

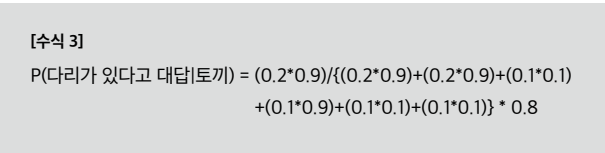
[표 2] 우도(Likelihood)

Likelihood	강아지	토끼	고래	호랑이	소라	뱀
다리가 있나요?	0.9	0.9	0.1	0.9	0.1	0.1
고양이와 크기가 비슷한가요?	0.8	0.9	0.1	0.2	0.1	0.2
바다에서 사나요?	0.1	0.1	0.9	0.1	0.9	0.2
알을 낳나요?	0.1	0.1	0.1	0.1	0.8	0.8
두 글자 인가요?	0.1	0.9	0.9	0.1	0.9	0.1

[표2]는 동물 스무고개에서 임의의 확률을 정한 표이다. 실 서비스에서 이 또한 [표1]과 같이 사용자 데이터로부터 갱신되어 확률이 변화하게 된다.

이제 준비가 끝났으니 사후분포를 구할 수 있다

'토끼'를 생각하고 있다고 가정하고 게임을 시뮬레이션 해보자. 질문의 순서 알고리즘은 배제하고 위의 다섯 문제를 순서대로 물어볼 것이다. 토끼의 사전 확률은 0.2이다. "다리가 있나요?"의 답은 '예'이므로 P(B|A) = 0.9를 곱한다. 식으로 나타내면 [수식 3]과 같다.



문제에 대한 답이 "아니오"인 경우에는 1-P(B|A)로 계산을 한다. 이렇게 다섯번째 문제까지의 토끼에 대한 대답으로 모든 동물의 사후 확률을 정리해 소수점 셋째 자리까지 표현해보면 다음과 같다.

[표 3] 사후확률(Posterior Probability)

Posterior	Answer	강아지	토끼	고래	호랑이	소라	뱀
다리가 있나요?	예	0.300	0.300	0.017	0.150	0.017	0.017
고양이와 크기가 비슷한가요?	예	0.351	0.395	0.002	0.044	0.002	0.005
바다에서 사나요?	아니오	0.353	0.398	0.000	0.044	0.000	0.004
알을 낳나요?	아니오	0.355	0.399	0.000	0.044	0.000	0.001
두 글자 인가요?	예	0.071	0.719	0.000	0.009	0.000	0.000

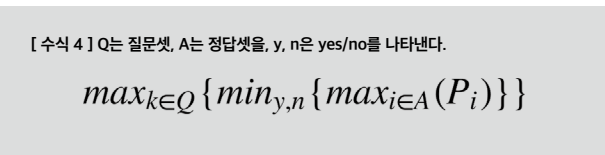
다섯 개의 질문을 한 끝에 사용자가 토끼를 생각하고 대답했을 확률이 약 72%로 다른 확률에 비해 월등히 높은 것을 볼 수 있다. 만약 "70% 이상이면 이 동물이 정답인지 물어보겠다"라고 설정해두었다면 다섯번째 질문 이후에 "토끼 인가요?"라는 질문이 나오고, '예'라는 대답이 돌아오면 게임이 끝나고 '아니오'가 돌아오면 토끼의 사후확률 값은 매우 적은 값, e로 치환되고 다음 질문들로 게임을 계속 진행한다.

"두글자인가요?"와 같이 답이 명백한 질문에도 잘못된 답을 할 가능성이 있어서 어떤 확률에도 극단적인 값인 0과 1은 발생하지 않는다. [표2]에 0.000으로 나타난 부분은 수가 매우 작아져서 소수점 셋째 자리로 잘랐을 때 뒤로 밀려나서이다.

다음 질문으로는 어떻게 좋을까

앞에서는 질문의 순서 알고리즘은 생각하지 않고 베이즈 정리만 써보았다. 데이터셋이 작을 때는 질문 순서를 랜덤으로 돌려도 20문제 안에 정답이 나올 확률이 크다. 하지만 데이터셋이 커지게 되면, 보다 효율적인 알고리즘을 필요로 한다.

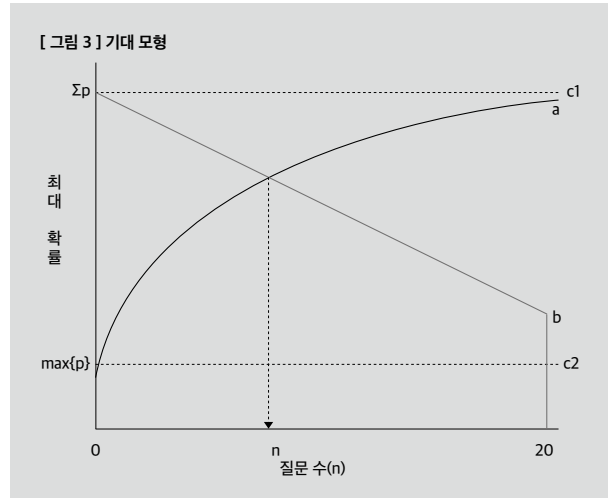
그렇다면 어떻게 다음 문제를 고를까. 스무고개 문제를 낼 때는 20개라는 한정된 질문만으로 답을 맞춰야한다. 보유하고 있는 질문셋에 대해 사용자가 '예/아니오(YES/NO)'로 대답한 이후의 각 정답셋에 대한 확률을 모두 계산한다. 그 중 우리가 선택하게 되는 질문은 각 질문에 대한 대답이 예(YES) 혹은 아니오(NO)일 경우의 사후 확률 중 낮은 값들을 비교해서 그 중 가장 큰 값을 고르는 것이다. 이를 식으로 나타내면 [수식 4]와 같다.



답을 제시하는 최적의 타이밍은

너무 빨리 답을 제시하면 정답률이 떨어지고, 늦게 제시하면 사용자가 지루함을 느끼게 된다. 따라서 질문의 수가 어느정도 일정하게 유지될 필요가 있다. 질문의 갯수가 적으면 보다 엄격한 기준을 제시하고, 많으면 느슨한 기준을 제시한다. 그리고 질문의 갯수는 특정값(예: 20)을 넘기지 않도록 한다.

기대하는 게임 모형과 앞으로의 과제



위의 스무고개 알고리즘 모델을 대략적으로 그려보면 이런 그림이 나온다. x, y축은 각각 게임에서 나온 질문의 수, 최대 사후 확률의 크기를 나타내며, a 곡선은 최대 확률곡선, b 선은 답을 제시할 최소 정답률을 나타낸다. a 곡선과 b선이 만나는 지점의 n번째(기댓값) 질문 이후 답을 제시하게 되는 것이다. 질문의 순서 알고리즘에 따라 a 곡선은 최대정답률에 빠르게 혹은 느리게 가까워진다. 즉, a 곡선이 b선과 만나는 n값이 작아지거나 커져서 평균 질문 수에 영향을 미치게 되는 것이다.

c1선의 윗 부분은 사용자가 데이터에 없는 대상을 생각한 경우, c2선은 가장 빈번하게 선택되는 대상이 정답일 확률값을 나타내며, 앞의 예시에서는 각각 0.8, 0.2이다.

스무고개의 퀄리티를 높이려면

보유 정답 데이터셋을 늘려서 최대 정답률을 높이거나 질문셋의 질을 높이고 순서를 적절히 배치한 후 정답에 이르는 질문의 수를 줄이도록 한다.

마치며

아직도 완벽하진 않지만 초기 버전의 스무고개는 답을 너무 못 맞춘다는 원성이 자자했다. 준비된 정답이나 질문의 개수도 한정적이고 확률 테이블도 구체적인 유저 데이터가 없어 추정치를 사용하고 있기 때문이라면 너무 구차한 변명인가? 수학적인 모형과는 별개로 인공지능 스피커 안의 게임이라고 해도 결국 사람이 만드는 거고 사람이 사용하는 것이기 때문에 '휴머니즘'을 완전히 배제하기가 어렵다. 인공지능 스피커를 사용하다보니 인간미가 느껴지는 부분이 킬링 포인트(killing point)인 것 같은데 이는 정답률에 비례해서 높아지는 것은 아닐 것이다.

너무 똑똑해도 밉다. 카카오미니와 함께하면 언제라도 부담없이 즐길 수 있는 적당히 똑똑하고 적당히 허술한, 내 친구 같은 게임 및 다른 형태의 인공지능 콘텐츠를 기대해본다.

“데이터는 모든 알고리즘을 압도한다.”

“数据秒杀算法”

- 바이두 CEO 로빈 리(李彦宏, 리옌홍) -

*1 참고 | California State University. "연령별 언어발달의 단계". <http://www.calstatela.edu/academic/ccoe/programs/cats-korean/연령별-언어발달의-단계> *2 참고 | 실제로는 "두 글자입니까?" 라는 간단한 질문에 도, 사람들은 "예"와 "아니오" 이외에도 "몰라", "글쎄-", "맞아!", "비밀이야", "안 가르쳐주지", "잘 아네", "3글자야" 등 다양하게 답하기는 한다. *3 참고 | <http://www.animalgame.com> *4 참고 | <https://www.techleer.com/articles/107-random-forest-supervised-classification-machine-learning-algorithm/> *5 참고 | <https://www.linkedin.com/pulse/significant-project-time-being-spent-data-preparation-wanjohi-ph-d/> *6 참고 | <http://www.20q.net> *7 참고 | <https://www.google.com/patents/US20060230008?dq=Artificial+neural+network+guessing+method+and+game> *8 참고 | <https://ecolloq.gsfc.nasa.gov/archive/2006-Spring/announce.burgener.html>, *9 참고 | https://ko.wikipedia.org/wiki/나이브_베이지_분류 *10 참고 | https://ko.wikipedia.org/wiki/베이지_추론