우도함수와 베이즈 정리를 중심으로 바라본 베이지안 최적화

수학과 소학회 MCS (Mathematics & Computer Science)

최보경, 이영진, 연성민, 이수정, 이윤지, 황유희

지도 교수 : 이영섭 교수님

베이지안 최적화란

베이지안 최적화는 실험 비용이 커서 여러 번의 실험을 진행할 수 없는 연구에서 주어진 데이터를 이용하여 함수를 추정하여 최댓값을 도출해낼 수 있는 입력값을 효율적으로 찾아내는 최적화 기법이다. 이때 최댓값이란, 어떤 문제를 해결할 수 있는 조건일때 나오는 최고의 결과이다.

이는 현재 기계학습, 로봇공학, 바이오/의약 분야 등 다양한 산업과 연구 분야에서 폭넓게 활용되고 있다.

베이지안 최적화 과정

1. Gaussian Process Regression

: 함수를 추정하기 위해 아래의 세 단계를 거친다.

1) mean function (**평균함수**)

: 함수 전체의 기본적인 형태를 초기 추정하는데 사용되는 것으로, 데이터를 보기 이전에 함수의 형태를 미리 추정한다.

일반적인 경우 : $\mu_0(x) = \mu$ * μ : 상수

추정하고자 하는 함수의 정보를 알 때 : $\mu_0(x) = \mu + \sum_{i=1}^{p} \beta_i \Psi_i(x)$

* μ : 상수 β_i : 회귀계수 $\Psi_i(x)$: 매개변수화된 함수

2) Kernel (**커널**) − Gaussian kernel (**가우시안 커널**)

: 커널 함수는 입력값 x x' 와 사이의 상관관계를 정의하는 함수로, 모델이 학습할 함수의 성질(부드러움, 주기성, 길이 척도 등)에 대한 사전적인 가정을 반영한다.
Gaussian Process에서는 공분산 함수의 역할을 하며, 관측되지 않은 점들의 함수값이 얼마나 비슷할지 표현한다.

$$\Sigma_0(x, x') = \alpha_0 \exp\left(-\|x - x'\|^2\right)$$

* α_0 : 분산의 크기를 조절하는 상수

* $\exp\left(-\|x - x'\|^2\right)$: 입력값 간 거리에 따라 공분산을 조절하는 항

3) Choosing a Hyperparameter (**하이퍼파라미터 추정**)

: 커널의 하이퍼파라미터들을 추정하는 과정으로,
이를 추정하기 위한 방법으로는 MLE와 MAP가 있다.
이때, 커널의 하이퍼파라미터에는 길이 척도, 분산 등이 있다.

2. Acquisition function (**획득함수**)

: 추정한 함수의 최댓값을 추정하기 위해 실험해볼 입력값들을 추려내는 단계이다.

우도함수와 MLE

1. 우도함수 (Likelihood)

: 주어진 데이터가 특정 모델에서 발생할 가능성으로,
임의의 하이퍼파라미터 η에 대해 관측 데이터 $f(x_{1:n})$가 나올 확률

$$P(f(x_{1:n}) \mid \eta)$$

* $f(x_{1:n})$: 관측된 데이터 η : 하이퍼파라미터 벡터

2. MLE (Maximum Likelihood Estimate)

임의의 하이퍼파라미터 η 에 대해 관측된 데이터 $f(x_{1:n})$ 가 나올 확률에서 가장 큰 확률을 갖도록 하는 하이퍼파라미터를 선택하는 것

$$\hat{\eta} = \arg\max_{\eta} P(f(x_{1:n}) \mid \eta)$$

베이즈 정리와 MAP

1. MAP (Maximum A Posteriori)

: 사후확률이 최대가 되도록 하는 η를 추정하는 방식으로,
우도함수에 사전확률을 곱해 사전정보를 반영함.

$$\hat{\eta} = \arg\max_{\eta} P(\eta \mid f(x_{1:n})) = \arg\max_{\eta} P(f(x_{1:n}) \mid \eta)\, P(\eta)$$

* $P(\eta \mid f(x_{1:n}))$: 사후확률 $P(\eta)$: 사전확률

2. 베이즈 정리 (Bayes' Theorem)

: 베이즈 정리를 이용하여 MAP 식을 아래의 과정을 통해 변형함.

$$P(\eta \mid f(x_{1:n})) = \frac{P(f(x_{1:n}) \mid \eta) \cdot P(\eta)}{\sum_{\eta'} P(f(x_{1:n}) \mid \eta') \cdot P(\eta')} \approx P(f(x_{1:n}) \mid \eta) \cdot P(\eta)$$

MLE와 MAP 비교

목표 : 자동차의 속도를 최대로 높일 수 있는 타이어의 압력 찾기

관측 데이터 : 타이어 압력과 자동차 속도에 대한 주행 데이터 (3회)
최대압력인 35psi에서 자동차의 속도가 최대가 됨

사전 정보 : 자동차 타이어의 압력이 35psi보다 높으면 속도 감소

1) MLE

: 오로지 관측 데이터만 고려하는 방법

→ 관측 데이터에서 최대 압력였던 35psi에서 자동차의 속도가 가장 빨랐기에, 압력이 증가할수록 자동차의 속도는 증가할 것

2) MAP

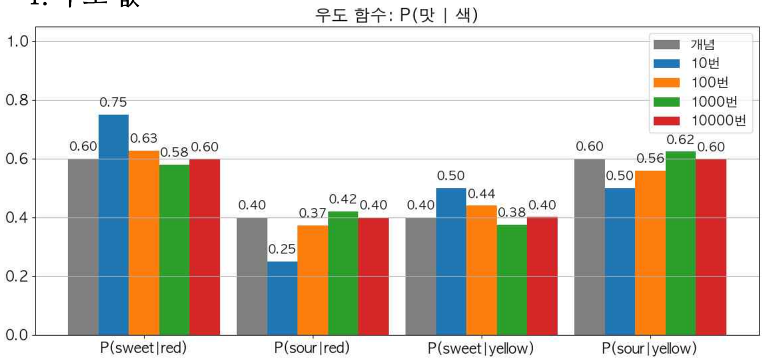
: 관측 데이터와 사전정보을 모두 고려하는 방법

→ 타이어의 압력이 35psi를 넘으면 속도가 감소하므로 압력이 35psi일때, 자동차는 최대속도를 낼 것이라고 예측

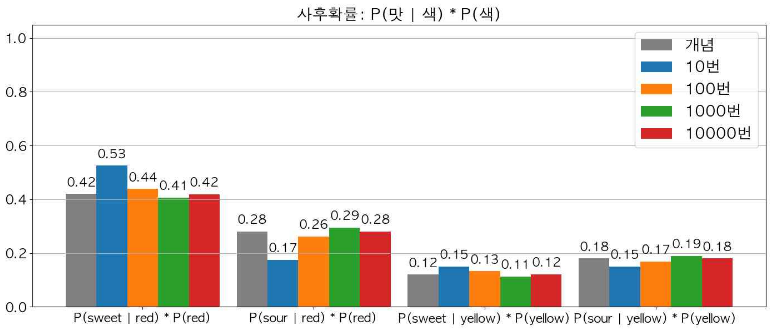
실제 모델로 증명하는 우도함수와 사후확률

종합 젤리 봉투 안에 아래의 비율로 젤리가 들어있다고 가정하자.
빨간색 젤리 : 노란색 젤리 = 7 : 3
빨간색 젤리 중 단 맛 : 신 맛 = 6 : 4
노란색 젤리 중 단 맛 : 신 맛 = 4 : 6

1. 우도 값



2. 사후확률 값



3. 결론

a. 시행횟수를 늘려 젤리를 10,000번까지 뽑았을 때, 실제 값과 이론적인 개념 값이 거의 같아지는 것을 확인할 수 있다.

b. 해당 사례에서는 우도 값보다 사후확률 값이 더 낮다는 것을 알 수 있다.

4. 첨부 – 코드

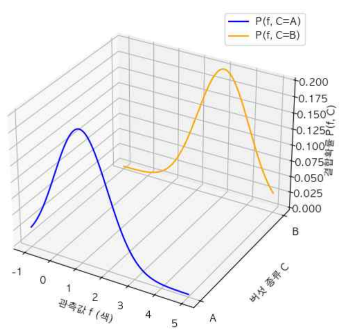
젤리는 파이썬 코드를 이용해서 뽑았다.

```python
import random
from collections import Counter # 결과 집계를 위함

# 젤리 종류와 확률, 딕셔너리 사용
jelly_types = {
    ("red", "sweet"): 0.42, ("red", "sour"): 0.28, ("yellow", "sweet"): 0.12,
    ("yellow", "sour"): 0.18
}

# 젤리 뽑기 시뮬 함수 정의
def simulate_jelly_draws(n):
    # 조합 리스트와 각 조합의 확률
    combinations = list(jelly_types.keys())
    probabilities = list(jelly_types.values())

    # 랜덤 뽑기
    results = random.choices(combinations, weights=probabilities, k=n)

    # 결과 집계
    counts = Counter(results)

    # 출력
    print(f"\n총 {n}번의 뽑기 결과:")
    for combo in combinations:
        color, taste = combo
        count = counts[combo]
        print(f"{color.capitalize()} & {taste.capitalize()}: {count}개")

if __name__ == "__main__":
    while True:
        try:
            n = int(input("\n뽑을 젤리 개수를 입력하세요 : "))
            if n > 0:
                simulate_jelly_draws(n)
            else:
                print("0 이상의 숫자를 입력해주세요.\n")
                continue
        except ValueError:
            print("숫자를 입력해주세요.\n")
```

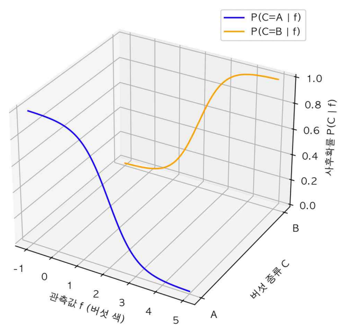
예시로 보는 우도함수와 사후확률

버섯의 밝기 정보를 이용하여 등산 중에 발견한 버섯이 독버섯인지 식용버섯인지 맞추어보자. 버섯의 밝기를 f라고 하고 버섯의 종류를 C라고 하자. 이때, 가정은 아래와 같다.
각 버섯 별로 하이퍼파라미터가 고정되어 있다고 가정하고, 이때의 하이퍼파라미터는 버섯의 종류를 의미하도록 한다.

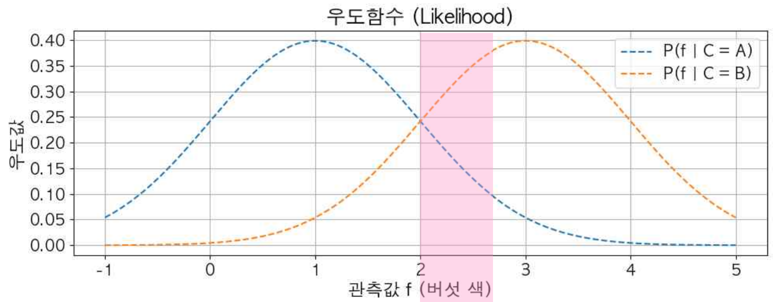
우도함수

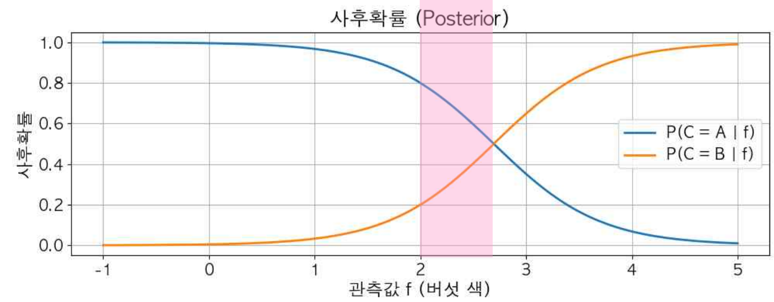


사후확률



비교를 위해
각각 3차원 형태인 우도함수와 사후확률의 그래프를
관측값 축에서 바라봤을때 얻어지는
평면 상의 그래프가 아래와 같다고 하자.





결론

관측값 f(= 버섯의 색)이 대략 [2, 2.69]일때
우도함수를 기준으로 보면, 발견한 버섯이 B일 확률이 높은데,
사후확률을 기준으로 보면, A일 확률이 높다.

이렇듯 우도함수로 판단하는 것과 사후확률로 판단하는 것의 결과는 다를 수 있다. 우도함수는 사전 정보를 반영하지 않기에 다소 극단적인 판단을 할 수 있고, 사후확률은 사전 정보를 반영하기에 보다 안정적인 판단을 할 수 있다.

참고문헌 : A Tutorial on Bayesian Optimization / Peter I. Frazier