함수모형 회귀분석 및 알고리즘

김석준*, 장근호**, 김예지*
*한신대학교 수학과,수리금융학과
e-mail : melon7607@naver.com

Function Regression algorithm

Seok Jun Kim*, Geun Ho Jang**, Ye Ji Kim*
*Dept of Mathematics , Han-Shin University
*Dept of Finance Mathmatical , Han-Shin University

요 약

Linear Regression 문제를 토대로 변형하여 선형회귀분석,2차함수모형 회귀분석, '단조 증가(감소)' 3차 함수 모형 회귀분석과 그에 따라 변형되는 gradient descent 알고리즘을 기술한다.

1. 서론

데이터를 분석하여 보여줄 때 모든 데이터를 '선형적 회귀분석' 으로만 분석하여 보여주어야 할까? 라는 생각으로 시작하여 LinearRegression 문제를 기반으로 선형,2차 함수 모형 회귀분석, '단조 증가(감소)' 3차 함수 모형 회귀분석 알고리즘을 구현 하였다.

회귀분석은 선형임을 가정하고 시작하지만 함수모형을 선정함에 따라 그 모형의 특징을 살릴 수 있기 때문인데 예를 들면 2차 함수 모형은 감소하다 증가하는 특징 ,'단조 증가(감소)'하는 3차 함수 모형은 증가하다 정체하게 되는 특징을 살릴 수 있다는 것이 이 논문의 논제이다.

2. Linear Regression

<표 1> 선형회귀

선형 회귀분석	
	두 데이터 값(x,y)간의 선형적 상관관계를 파악한다. 예시) 경력에 따라 시급이 오를까? =>분석시 함수 f(x)로 도출 =>$f(x)=30x+270$ 존재하지 않는 데이터도 함수로 추측가능 => 3개월 차 월급 $f(3)=30*30+270$ $=90+270=360$(만)
일반적인 선형회귀분석을 기계학습을 통해 구현 할 수 있다.	

Linear Regression 문제를 말한다.

통계학적으로는 '선형 회귀분석'의 모델을 컴퓨터가 학습할 수 있도록 설계하여 상수 값을 갖는 두 개 이상의 변수를 갖는 데이터가 주어지면 그 변수간의 상관관계를 학습시켜 가장 적합한 하나의 직선을 찾아내는 문제이다.

여기서 가장 적합한 하나의 직선이 되는 기준은 cost라는 비용으로 함수와 데이터 간의 차이(편차)의 제곱의 평균 (분산과 비슷하다 생각하면 된다.)이 가장 작을 때의 함수가 가장 적합하다는 기준이 되며 gradient descent알고리즘을 통해 적합 함수를 찾아낸다.

3. 2차 함수 모형과 회귀분석

<표 2> 2차 함수 모형

2차 함수모형 분석	
	어떤 데이터가 증가하다(+) 감소(-)또는 감소하다(-) 증가(+) 하는 데이터라면 2차 함수모형이 적합하다 $f(x)=w(x-x_1)^2+y_1$ 존재하지 않는 데이터도 함수로 추측 가능하다
선형회귀를 변형하여 2차 함수모형으로 기계학습을 시켜 분석한다.	

2차 함수모형 알고리즘 위에서 '선형적'이란 단어의 뜻은 그래프 관점에서 보면 '직선적인 관계'로 볼 수 있다. 그러나 모든 데이터의 분포를 선형적으로만 분석하게 된다면 문제가 있다고 보았다. 따라서 비선형적인 모델로 분석하기 위해 정의역이 주어진 전사함수를 선정하게 되었고 가장 일반적인 함수들을 채택하게 되었다.

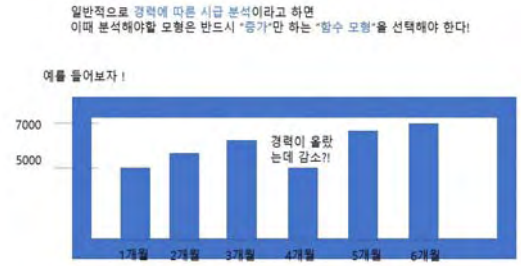
3. '단조 증가(감소)' 3차 함수 모형 3차 함수 모형과 회귀 분석

<표 3> '단조 증가(감소)' 3차 함수 모형

'단조 증가(감소)' 3차 함수 모형	
	어떤 데이터가 '정체 구간'을 표현하기에 적합하다. 예시) 경력에 따라 수익이 오르지 않는 구간도 생길 수 있으며 계속적으로 증가하는 함수이므로 분석 시 감소하는 구간이 있으면 안 되는 데이터에 적합하다.
선형회귀를 변형하여 '증가(감소) 또는 정체' 만 하는 3차 함수 모형으로 기계학습을 시켜 분석한다.	

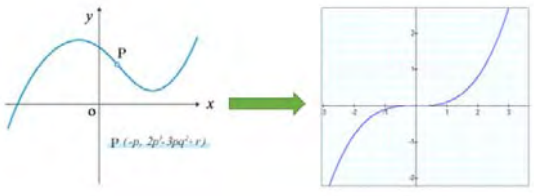
여기서 3차 함수를 '단조 증가(감소)' 3차 함수 모형으로 변형하고 선정한 이유는 이 함수만의 특징이 있기 때문이다.

먼저 앞서 설명하면 경력에 따른 월급분석을 할 때 경력이 오르고 있는데 월급이 떨어지게 되는 구간이 있다는 것은 일반적으로 분석이 잘못 됐다고 생각하여 신뢰도가 떨어지게 될 것이다.



(그림 1) 분석결과가 단조증가 해야 하는 예시

따라서 원래 3차 함수모형은 아래와 같이 변형하여 수식을 변형하였다.



(그림 2) 일반 3차 함수에서 '단조 증가(감소)' 3차 함수 모형으로의 변형

즉 이 함수모형은 어떤 증가하다가 '정체되는 구간'을 표현하기 좋은 모형이며 시간복잡도 또한 수정할 weight 값이 적어 실시간 응답에 적합한 함수모형이다.

4. 함수 모형 회귀분석 알고리즘

<표 4> 함수 모형 회귀분석

함수모형 회귀분석(제네릭 버전)	
	앞의 데이터들을 각각 데이터의 분포에 알맞은 정도(cost)로 기준을 두고 가장 분포에 알맞은 함수를 채택하여 보여준다.
데이터들을 선형 , 2차 모형 ,단조 증가(감소) 3차 모형들로 각각 분석해 보고 그 중에서 데이터 분포에 가장 적합하다고 판정되는 것으로 출력한다.	

함수모형 알고리즘은 위에서 기술한 모형들로 각각 분석해 보고 그중에서 분포에 가장 알맞은 정도 (cost)를 기준으로 분포에 가장 알맞은 함수를 채택하는 것을 말한다.

<표 5> 함수 모형 회귀분석의 원리

	함수식	cost
LinearRegression	$f(x)=w(x-a)^1+b$	100
2차함수 모형	$f(x)=w(x-a)^2+b$	3000
단조 3차 함수 모형	$f(x)=w(x-a)^3+b$	80
...	추가되는 함수모형들..	..
function Regression (함수 모형 회귀분석)	단조 3차 함수 모형이 cost값이 가장 작음 = 단조 3차 출력	80

5. 선형,2차,단조 증가(감소) 3차 모형 알고리즘

위의 알고리즘을 구현하기 위해서는 크게 다음과 같은 순서로 진행된다.

1) 데이터를 1,2차 단조 3차 함수 모형이라고 각각 가정하고 그에 따라 추측되는 중심 좌표
(알고리즘 상 centerPoint)를 찾아낸다.

2) 그 좌표를 함수식의 (a,b) 값에 대입하여 weight값을 수정해 나간다. (gradient descent 알고리즘)

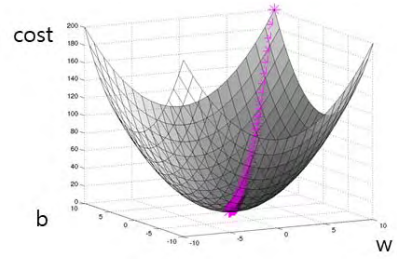
3) function Regression일 경우 각 모형들을 분석했을 때 가장 cost(비용)값이 작은 함수모형을 채택하여 보여준다.

먼저 1)의 내용은 각 함수식을 표준형으로 추상해야한다는 것에 중점을 둔다.

<표 6> 함수의 표준형의 중요성

일반형	$f(x)=w_1x^2+w_2x+w$	수정해야할 weight값이 많다. (3개)
표준형	$f(x)=w(x-a)^2+b$	꼭지점 좌표를 대략 찾아 수정 할 weight값을 1개로 줄인다.

일반형으로 분석하게 되면 물론 결과 함수는 더욱 정확하겠지만 이러한 알고리즘 방식으로 작동하면 시간 복잡도나 그래프를 추상하기 매우 간단해 진다.
예를 들면 y=wx+b의 형태의 단순한 모형도 따지고 보면 조정해야할 변수가 2개이다 그런데 gradient descent알고리즘을 사용하기에 적합한지 확인하고자 하면 전체적인 그래프를 그려보게 되는데 조정해야할 변수가 2개이고 cost값을 y축에 세워보면 결국 3차원의 곡면이 최소 cost값으로 수렴할 수 있도록 그래프가 이루어 져야한다.



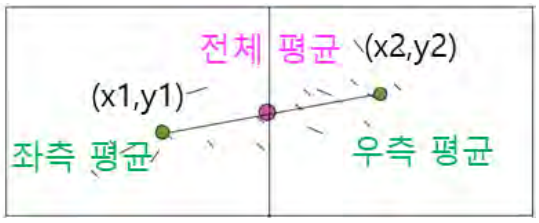
(그림 3)조정할 weight 값을 1개로 줄이는 이유
(y=wx+b의 예시)

3차원 까지는 무난하다 하자 그런데 2차 함수 모형이나 '단조 증가(감소)' 3차 함수 모형에서는 조정해야할 변수가 3개가 되는데 이를 gradient descent알고리즘을 사용하기 적합한지 확인하려면 4차원을 생각하게 되므로 가능하다 하더라도 추상하기 어려움이 있다.

따라서 여기서는 어떤 centerPoint(a,b) 좌표를 두어 모형별로 "어떤 좌표를 대략 지날 것인 가?"를 통계적 방식으로 먼저 추측하고 그 좌표를 상수값으로 대입하여 하나의 weight만을 수정하겠다는 방식으로 구성되었다.

따라서 표준형으로 함수식을 설계하고 그에 따른 중심좌표(a,b)를 미리 구하는 것이 매우 중요한데 이는 다음과 같다.

1-1) Linear Regression(선형회귀) 의 centerPoint(a,b)

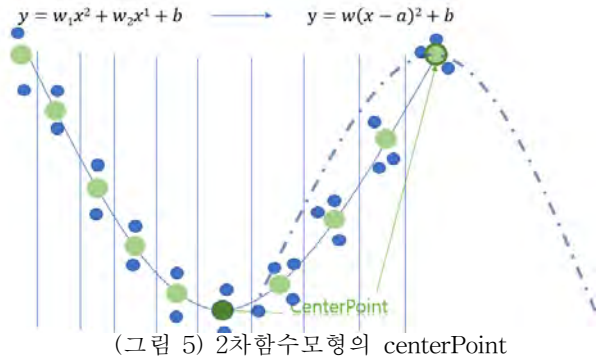


(그림 4) 선형회귀모형의 centerPoint와 초기 weight

그림에서 centerPoint는 빨간색 좌표를 뜻한다.
즉 데이터들을 선형으로 분석한다는 것은 데이터의 분포가 선형적으로 분포 되어있다고 가정하고 시작할 수 있으므로 "선형적으로 분포되어 있을 때 구하고자 하는 회귀직선은 어떤 좌표를 지나게 될까?" 의 대답은 전체 데이터들의 x,y값의 평균 값을 반드시는 아니어도 centerPoint(a,b)로 선정하기에는 무리가 없다고 생각하였다.

또한 위에서 centerPoint를 기준으로 좌측 평균,우측 평균도 추가로 구하였는데 이것의 효과는 초기의 weight 값을 대략적으로 추상하여 기울기(w)의 초기값을 적절하게 선정하기 위함이다. ($w = \frac{y_2 - y_1}{x_2 - x_1}$)

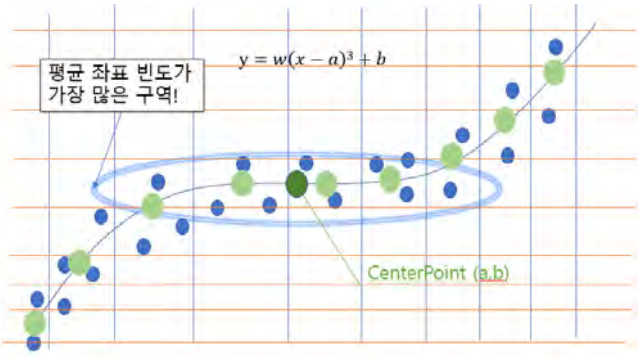
1-2) 2차 함수 모형의 centerPoint(a,b)



(그림 5) 2차함수모형의 centerPoint

"2차 함수 모형으로 분포되어 있을 때 구하고자 하는 회귀 포물선은 어떤 꼭지점 좌표 위치를 지나게 될까?"
라고 출발하여 적절한 centerPoint좌표를 찾아낸다.
1. 데이터들의 최소 x값 , 최대 x값을 구한 뒤 그 범위를 10등분하고 등분된 부분의 평균 포인트들을 구한다.
2. 양수일 경우 : 10개의 평균 Point들의 최소 Point
음수일 경우 : 10개의 평균 Point들의 최대 Point
3. 구해진 2개의 좌표로 gradient descent알고리즘을 각각 실행하여 둘 중 cost값이 낮은 centerPoint와 weight를 가져온다.

1-3) '단조 증가(감소)' 3차 함수 모형 centerPoint



(그림 6) 3차 함수 모형의 centerPoint

"'단조 증가(감소)' 3차 함수 모형"으로 분포되어 있을 때 구하고자 하는 회귀선은 어떤 좌표를 중심으로 지나게 될까?" 라고 출발하여 적절한 centerPoint좌표를 찾아낸다.
1. X (Key)의 최소 값과 최댓 값으로 x축 10등분
2. 10등분된 각 범위별 평균 좌표를 찾아냄
3. Y (Value)의 최솟값과 최댓값으로 y축도 10등분
4. 등분된 구역중 2에서 얻어낸 평균 좌표의 빈도가 가장 많은 부분을 찾는다
5. 그 구역의 평균좌표가 CenterPoint 이다.
이렇게 채택된 centerPoint로 시작하여 gradient descent 알고리즘을 하게 된다.

2) 함수 모형에 따른 gradient descent 알고리즘

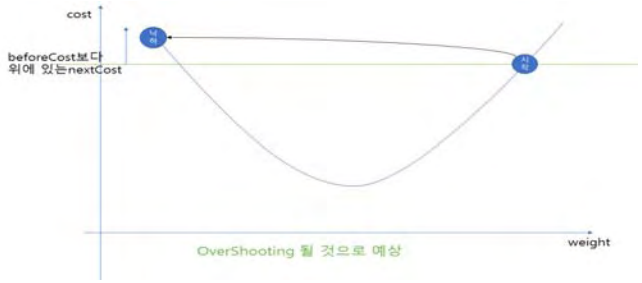
<표 7> 함수모형식과 gradient descent 함수식

	모형 함수 식	gradient descent
1차 함수 모형	$f(x)=w(x-a)^1+b$	$w=w-\frac{r}{n}\sum_{i=1}^{n}(w(x_i-a)^1+b-y_i)(x_i-a)^1$
2차 함수 모형	$f(x)=w(x-a)^2+b$	$w=w-\frac{r}{n}\sum_{i=1}^{n}(w(x_i-a)^2+b-y_i)(x_i-a)^2$
단조 3차 함수 모형	$f(x)=w(x-a)^3+b$	$w=w-\frac{r}{n}\sum_{i=1}^{n}(w(x_i-a)^3+b-y_i)(x_i-a)^3$

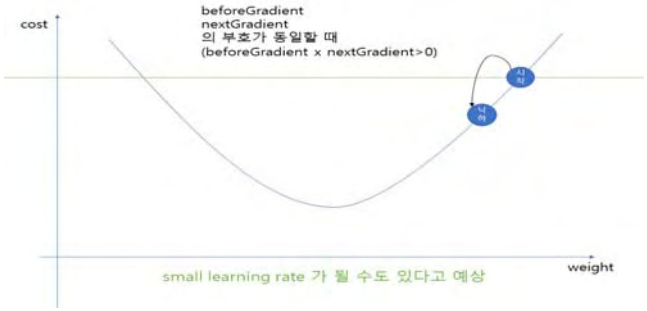
함수식이 변형됨에 따라 gradient descent 알고리즘의 함수식 또한 cost 함수를 w에 대해 편미분 시켜 변형된다.

여기서 중요한 부분은 learning rate라고 불리는 함수식에서의 r값인데 이 값이 너무 높으면 overShooting현상 또는 small rate현상이 일어나기 때문에 적절한 학습계수를 선정하는 것이 중요하다.

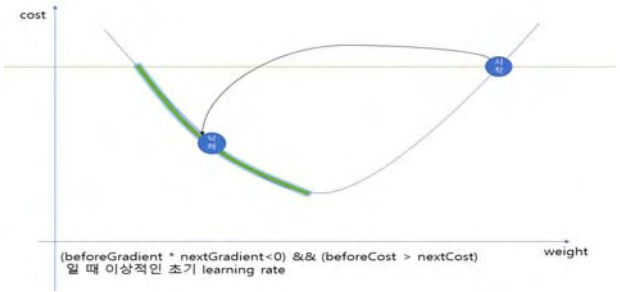
필자는 다음과 같은 방식으로 학습계수를 조절하였다.



(그림 7) overShooting 예상 범위



(그림 8) small learning rate 예상 범위



(그림 9) 이상적인 초기 learning rate 선정 범위

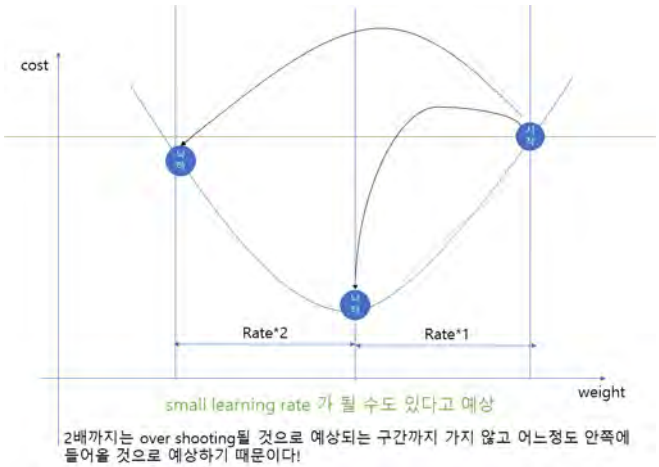
여기서 하고자하는 learning rate 조절 방식은 "초기값"을 잘 선정하여 이상적인 learning rate값을 찾아낸 뒤 overShooting 또는 small rate 현상이 일어날 것으로 보이면 learning rate값을 조금씩 늘리거나 줄여나가 학습시키자는 방식이다.

```
rate = toInitLearningRate(w,centerPoint,0.01,degree);
// learningRate값 초기값 (초기화)
for (int i = 0; i < randomCount; i++) {

.... //생략
if(beforeCost<afterCost) {
//OverShooting 예상됨 초기화 된 rate값을 줄인다.
                rate*=0.1;
}else if(beforeGradient*afterGradient>0) {
// learning rate 올려도 됨. (*꼭 2배 이하)
                rate*=2;
                }
            if(beforeCost>afterCost) {
              resultW = w;
              beforeCost = afterCost;
}
          if(afterGradient==0) {
break;
                }
          }
return resultW;
```

*rate값을 늘릴 때는(2배 이하로 늘려야 한다.)

beforeGradient*afterGradient>0 의 의미는 참고로 기울기의 부호가 서로 같다는 것을 표현하기 위해 표현하였다. 이때 rate 값을 1배보다는 크고 2배 이하로 조절하게 되는데 이유는 다음과 같다.



(그림 10) rate 값을 늘릴시 1~2배 이하로 선정해야하는 이유

2배까지는 overShooting이 될 것으로 예상되는 구간까지 가지 않고 어느정도 안쪽에 들어올 것으로 예상하기 때문에 안정적이라 생각하였다.

따라서 함수모형 회귀분석 알고리즘을 구현하였으며 "한이음 알바 셀파 프로젝트"에 적용하고 구현시켰다.

참고문헌

[1] 모두를 위한 머신러닝 강좌
http://hunkim.github.io/ml/ 김성훈 저
[2] 쉽게 배우는 알고리즘 (관계 중심의 사고법)
문병로 저