11. 최적화 알고리즘

- 패턴 인식은 최적화 문제를 많이 다룬다.
  - SVM은 여백을 최대로 하는 결정 직선을 찾는다.
  - 퍼셉트론이나 MLP는 오류를 최소로 하는 가중치 값을 찾는다.
- 일반적으로 최적화 문제는 (11.1)과 같이 기술할 수 있다.

최대화 문제

$J(\theta)$를 최대로 하는 $\hat{\theta}$를 찾아라. 즉, $\hat{\theta} = \arg\max_{\theta} J(\theta)$ 이다.

$J(\theta)$를 최소로 하는 $\hat{\theta}$를 찾아라. 즉, $\hat{\theta} = \arg\min_{\theta} J(\theta)$ 이다.

패턴 인식의 문제 풀이는 크게 두 단계로 나누어 볼 수 있다.

1단계

최적화 문제로 공식화 (즉 $J(\theta)$를 정의함)

↓

2단계

$J(\theta)$를 최대 또는 최소로 하는 $\hat{\theta}$를 찾아라.

즉, $\hat{\theta} = \arg\max_{\theta}(\theta)$ or $\hat{\theta} = \arg\min_{\theta}(\theta)$

- 11장의 목적
  - 이미 1-10장에서 배운 최적화 알고리즘을 보다 명시적으로 드러내고 그들을 비교함으로써 알고리즘에 대한 이해의 깊이를 더함
  - 새로운 최적화 알고리즘 소개
    (시뮬레이티드 어닐링, 유전 알고리즘)

11.1 패턴 인식의 최적화 문제 풀이

- 분류기 학습
  - 학습 집합 $X = \{(x_1, t_1), (x_2, t_2), \cdots, (x_N, t_N)\}$
  - θ는 가중치 벡터

$$J(\theta) = \sum_{i=1}^{N} d(o_i, t_i)$$

그림 11.2 분류기 학습 문제

• 선택

- 특징 선택 (9장), 분류기 앙상블 선택(12장), k-NN을 위한 프로토타입 선택 등
- 선택 문제도 θ를 도입하여 표현할 수 있다. θ는 n비트를 갖는 이진열이다. 예를 들어 $n=5$인 경우 부분 집합 $S_1=\{s_2,s_3,s_5\}$는 $\theta=01101$로 표현한다. 즉 선택된 요소는 1을 갖고 제거된 요소는 0을 갖는다.

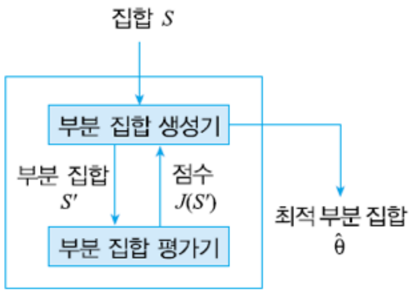


그림 11.3 선택 문제

• 군집화

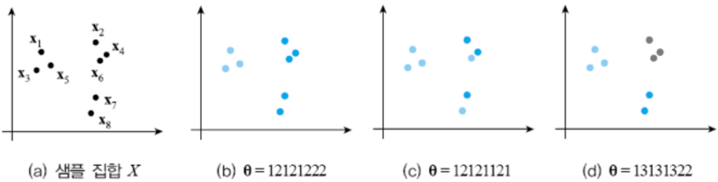


그림 11.4 군집화 예 (연한 파랑은 군집1, 진한 파랑은 군집2, 검정은 군집3)

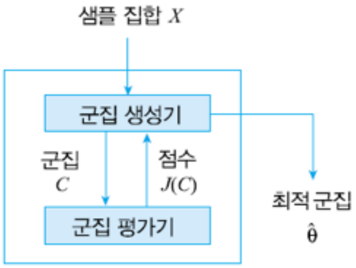


그림 11.5 군집화 문제

11.1.2 문제풀이

이들 문제를 어떻게 풀어 최적의 매개 변수 값을 찾아 낼 것인가?

- 문제의 난이도
  - θ의 차원. 작게는 수십~수백, 크게는 수천~수만 차원
  - 목적 함수 J(θ)의 복잡도. 지역 최적점이 하나 뿐인 Single-modal인 경우 미분식을 풀어 해결 가능, multi-modal인 경우는 보다 복잡한 알고리즘 필요
- 해 공간을 효율적으로 탐색하는 알고리즘 필요

- 문제 풀이
  - 분석적 방법: 미분식을 풀어 해를 구함
  - 수치적 방법: 초기 해에서 출발하여 그것을 조금씩 개선해 나가는 방법

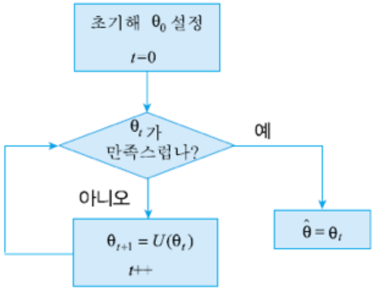


그림 11.6 최적해를 찾기 위한 반복 알고리즘

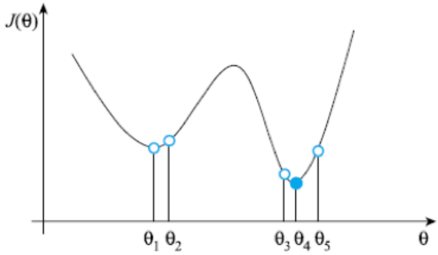


그림 11.7 최소화 문제에서 최적 해 (속이 찬 점)와 부 최적 해 (속이 빈 점)

최적해(전역 최소점, θ_4)을 보장하는 것은 아니다. 지역 최소점 (θ_1) 또는 최소 점 근방($\theta_2, \theta_3, \theta_5$)을 찾고 멈출 수도 있다.

11.2 미분을 이용한 방법

11.2.1 분석적 풀이

알고리즘 [11.1] 분석적 풀이

입력: 목적 함수 $J(\boldsymbol{\theta})$

출력: $\hat{\boldsymbol{\theta}}$ (최고 점 또는 최저 점)

알고리즘:

1. $J(\boldsymbol{\theta})$를 $\boldsymbol{\theta}$로 미분한다.
2. 방정식 $\dfrac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$ 을 만족하는 $\hat{\boldsymbol{\theta}}$ 를 구한다.
3. **return** $\hat{\boldsymbol{\theta}}$;

예제 11.1 미분에 의한 분석적 방법

■ 두 가지 예제

(1) 그림 11.8(a): $J(\theta) = -2\theta^2 + 20\theta - 43$

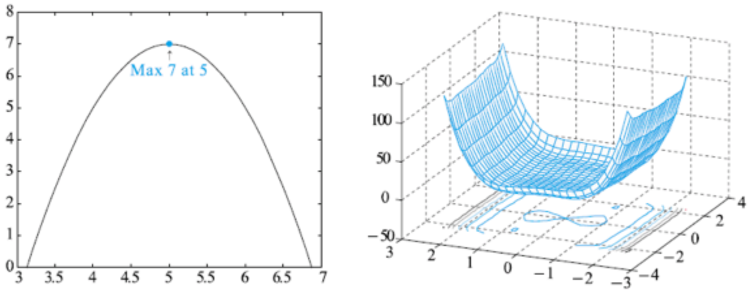
$\frac{\partial J}{\partial \theta} = -4\theta + 20 = 0$ 을 풀면 $\hat{\theta} = 5$ 이다.

(2) 그림 11.8(b): $J(\boldsymbol{\theta}) = (4 - 2.1\theta_1^2 + \theta_1^4/3)\theta_1^2 + \theta_1\theta_2 + (-4 + 4\theta_2^2)\theta_2^2$

$\frac{\partial J}{\partial \boldsymbol{\theta}} = (\frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2})^T = (2\theta_1^5 - 8.4\theta_1^3 + 8\theta_1 + \theta_2, 16\theta_2^3 - 8\theta_2 + \theta_1)^T = \mathbf{0}$ 을 풀면

$\hat{\boldsymbol{\theta}} = (-0.0898, 0.7126)^T$ 또는 $\hat{\boldsymbol{\theta}} = (0.0898, -0.7126)^T$ 에서 최소값 -1.0316을 갖는 것

알 수 있다.



(a) 이차 함수의 최대화

(b) 여섯 혹을 가진 낙타 등 함수의 최소화

그림 11.8 미분에 의한 최적해 구하기

11.2.2 내리막 경사법

• 해를 반복 개선하는 수치적 방법의 일종

– 현재 위치에서 경사가 가장 급격하게 떨어지는 방향을 찾고 그 방향으로 해를 약간 이동. 방향은 도함수로 알아냄

– ρ는 학습률로서 이동량을 조절

내리막 경사법 $\theta_{t+1} = \theta_t - \rho \left. \frac{\partial J}{\partial \theta} \right|_{\theta_t}$

오르막 경사법 $\theta_{t+1} = \theta_t + \rho \left. \frac{\partial J}{\partial \theta} \right|_{\theta_t}$

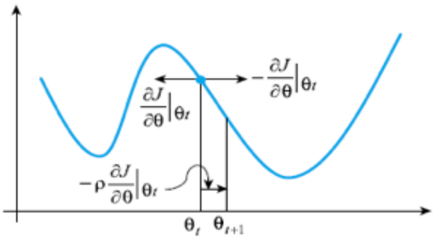


그림 11.9 내리막 경사법

예제 11.2 내리막 경사법 예제 11.1의 낙타 등 함수

- 도함수는 $J'(\boldsymbol{\theta}) = \frac{\partial J}{\partial \boldsymbol{\theta}} = (\frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2})^T = (2\theta_1^5 - 8.4\theta_1^3 + 8\theta_1 + \theta_2, 16\theta_2^3 - 8\theta_2 + \theta_1)^T$
- 초기값을 $\boldsymbol{\theta}_0=(-0.5,0.5)^T$, 학습률 $\rho=0.01$로 하면,

$$\frac{\partial J}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}_0} = (-2.5125, -2.5)^T$$

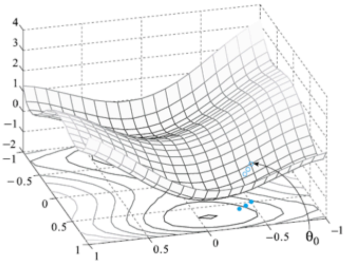
$$\boldsymbol{\theta}_1 = \boldsymbol{\theta}_0 - 0.01 * \frac{\partial J}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}_0} = (-0.5, 0.5)^T - 0.01 * (-2.5125, -2.5)^T = (-0.4748, 0.525)^T$$

$$\frac{\partial J}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}_1} = (-2.4228, -2.3596)^T$$

$$\boldsymbol{\theta}_2 = \boldsymbol{\theta}_1 - 0.01 * \frac{\partial J}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}_1} = (-0.4748, 0.525)^T - 0.01 * (-2.4228, -2.3596)^T$$
$$= (-0.4506, 0.5486)^T$$

점점 낮은 곳을 찾아간다.

$$J(\boldsymbol{\theta}_0) = -0.12604$$
$$J(\boldsymbol{\theta}_1) = -0.24906$$
$$J(\boldsymbol{\theta}_2) = -0.36036$$



11.2.3 라그랑제 승수

조건부 최적화 예 $J(\theta) = \theta_1^2 + 2\theta_2^2$

- 최소점 $(0,0)^T$
- 하지만, '$2\theta_1 + \theta_2 = 0$ 이라는 조건 하'에 최소점을 구하는 문제라면?

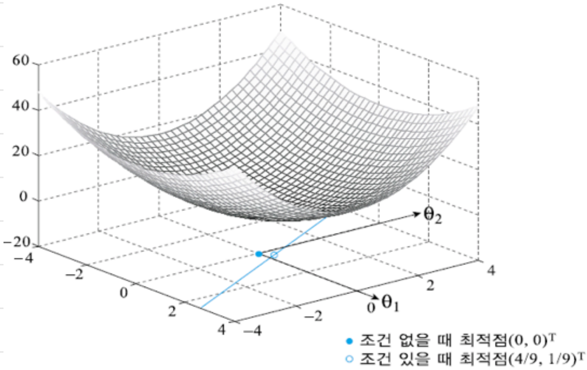


그림 11.11 등식 조건부 최적화 문제의 예

- 등식 조건부 최적화 문제 (equality constrained optimization problem)
  - 등식 조건부 최소화 문제

$$\left(\begin{array}{l} \text{아래 조건하에,} \\ f_i(\theta) = 0, \quad i = 1, \cdots, n \\ J(\theta)\text{를 최소화하는 해 } \hat{\theta}\text{를 구하라.} \end{array} \right) \quad (11.5)$$

• 라그랑제 함수와 라그랑제 승수

- (11.6)은 라그랑제 함수
- 조건식마다 라그랑제 승수 λ_i 를 할당

$$L(\theta, \lambda) = J(\theta) \underbrace{-}_{\text{최대화일 경우 } +} \sum_{i=1}^{n} \lambda_i f_i(\theta)$$

• 이제 라그랑제 함수를 최소화 하는 해를 구하면 된다!

결국 (11.7)을 만족하는 해를 구하면 됨

$$\left.\begin{aligned} &\frac{\partial L}{\partial \theta_1} = 0, \quad i=1, \cdots, k \\ &\frac{\partial L}{\partial \lambda_i} = 0, \quad i=1, \cdots, n \end{aligned}\right) \qquad (11.7)$$

예제 11.3 등식 조건부 최적화 문제

• $2\theta_1 + \theta_2 = 1$ 이라는 조건 하에, $J(\theta) = \theta_1^2 + 2\theta_2^2$ 의 최소점을 구하라.

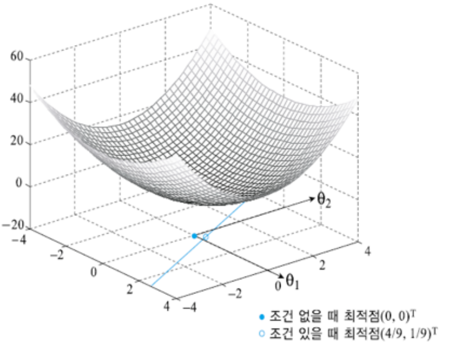


그림 11.11 등식 조건부 최적화 문제의 예

$$L(\theta, \lambda_1) = (\theta_1^2 + 2\theta_2)^2 - \lambda_1(2\theta_1 + \theta_2 - 1)$$

θ로 미분한 식을 0으로 둠 : $\frac{\partial L}{\partial \theta_1} = 2\theta_1 - 2\lambda_1 = 0$

$\frac{\partial L}{\partial \theta_2} = 4\theta_2 - \lambda_1 = 0$

λ로 미분한 식을 0으로 둠: $\frac{\partial L}{\partial \lambda_1} = -(2\theta_1 + \theta_2 - 1) = 0$

$$\Rightarrow \hat{\theta} = \left(\frac{1}{9}, \frac{4}{9}\right)^T$$

- 부등식 조건부 최적화 문제

아래 조건하에,

$f_i(\theta) \geq 0 \ , \ i=1, \cdots, n$

$J(\theta)$를 최소화하는 해 $\hat{\theta}$를 구하라 (11.8)

이 문제는 Karush-Kuhn-Tucker(KKT) 조건을 사용하여 푼다.

$$\frac{\partial L(\theta, \lambda)}{\partial \theta} = 0$$

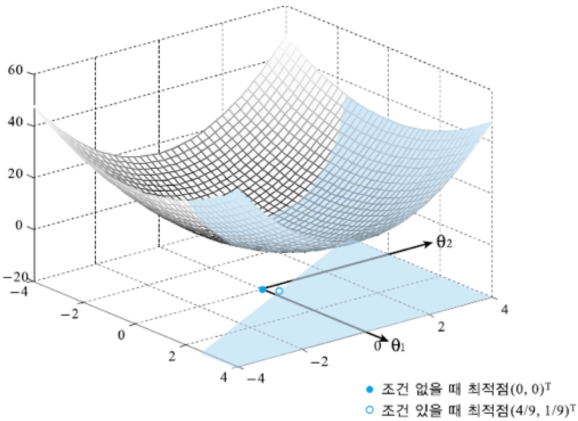
$$\lambda_i \geq 0, \ i=1, \cdots, n$$

$$\lambda_i f_i(\theta) = 0, \ i=1, \cdots, n$$

KKT 조건

예제 11.4 부등식 조건부 최적화 문제

- 그림 11.11을 부등식 조건으로 바꾸어 보면,
  - '$2\theta_1+\theta_2>1$이라는 조건 하'에 $J(\boldsymbol{\theta}) = \theta_1^2+2\theta_2^2$ 의 최소점을 구하라.



(a) $f_1(\boldsymbol{\theta}) = 2\theta_1 + \theta_2 - 1 \geq 0$

KKT 조건을 구하고 풀면,

$$L(\theta,\lambda) = (\theta_1^2+2\theta_2^2) - \lambda_1(2\theta_1+\theta_2-1)$$

$$\frac{\partial L}{\partial \theta_1} = 2\theta_1 - 2\lambda_1 = 0$$

$$\frac{\partial L}{\partial \theta_2} = 4\theta_2 - \lambda_1 = 0$$

$$\lambda_1 \geq 0$$

$$\lambda_1(2\theta_1+\theta_2-1) = 0$$

① $\lambda_1 = 0$

$\theta_1=\theta_2=0$, $f_1(\theta) = 2\theta_1+\theta_2-1 \geq 0$을 만족하지 못함

② $2\theta_1+\theta_2-1=0$

$$\Rightarrow \hat{\theta} = \left(\frac{4}{9}, \frac{1}{9}\right)^T$$

표 11.1 패턴 인식에 나타난 다양한 최적화 문제들과 그들이 사용한 최적화 알고리즘

최적화 문제	매개 변수 θ	문제 풀이	관련 수식	알고리즘
최대 우도 (3.2절)	pdf의 매개 변수	분석적 방법	문제 (3.4) 해 (3.6)	수식 (3.6)을 계산 (pdf가 가우시언일 때)
GMM (3.4절)	(μ_k, Σ_k), π_k $1 \leq k \leq K$	분석적 방법과 라그랑제 승수, 내리막 경사법	문제 (3.23) 해 (3.26), (3.28), (3.29)	알고리즘 [3.3] (EM 알고리즘)
퍼셉트론 학습 (4.2절)	$\mathbf{w}$	내리막 경사법	문제 (4.4) 해 (4.7)	알고리즘 [4.1]
다층 퍼셉트론 학습 (4.3절)	$\mathbf{u}$, $\mathbf{v}$	내리막 경사법	문제 (4.16) 해 (4.18), (4.19), (4.20), (4.21)	알고리즘 [4.5] (오류 역전파 알고리즘)
SVM 학습 (5장)	$\mathbf{w}$	라그랑제 승수	문제 (5.13), (5.27), (5.35) 해 SMO 방법 등	알고리즘 [5.1]
PCA (8.5절)	고유벡터	라그랑제 승수	문제 (8.32) 해 (8.33)	알고리즘 [8.3]
특징 선택 (9.4절)	부분집합	내리막 경사법	문제 (9.5) 해 (9.7), (9.8)	알고리즘 [9.5]~[9.8]

11.3 시뮬레이티드 어닐링

- 시뮬레이티드 어닐링 기본 원리
  - 내리막 경사법과 비슷함
  - 해를 하강시키기만 하는 경사법과 달리, 조건에 따라 해를 상승하기도 함 (이 기능으로 지역 최적점 탈출하는 여지 생김)
  - 온도라는 뜻을 갖는 변수 T를 조절함으로써 상승 확률 조절 (초기에는 온도가 높아 상승 확률 크지만 시간이 지남에 따라 온도를 낮추어 확률을 줄임)

알고리즘 [11.3] 시뮬레이티드 어닐링

입력: 목적 함수 $J(\boldsymbol{\theta})$

출력: 최적해 $\hat{\boldsymbol{\theta}}$

알고리즘:

이웃기준? 비트 수 1개 허용하면

특징선택에서, θ = 0111001011

1111001011 (이웃) → 10개의 이웃 존재,

비트 수 2개 허용 → 45개의 이웃 허용

1. 초기해 $\boldsymbol{\theta}$를 설정한다.
2. $score = \text{goodness}(J(\boldsymbol{\theta}))$;
3. 초기 온도 T를 설정한다.
4. $best\_score = 0$;
5. **repeat** {
6. $\quad \boldsymbol{\theta}_{new} = \text{random\_neighbor}(\boldsymbol{\theta})$;
7. $\quad new\_score = \text{goodness}(J(\boldsymbol{\theta}_{new}))$;
8. $\quad$ **if** (random() < accept_level($score$, new_score, T))
   $\quad\quad$ { $\boldsymbol{\theta} = \boldsymbol{\theta}_{new}$; $score = new\_score$;}
9. $\quad$ **if** ($new\_score > best\_score$) // 새로운 최적 점 발견
   $\quad\quad$ { $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{new}$; $best\_score = new\_score$;}
10. $\quad$ 온도 T를 낮춘다.
11. } **until** (stop-condition);
12. **return** $\hat{\boldsymbol{\theta}}$;

13. accept_level($s1$, $s2$, T) {
14. $\quad$ **if** ($s2 > s1$) **return** 1;
15. $\quad$ **else return** $e^{-(s1-s2)/T}$;
16. }

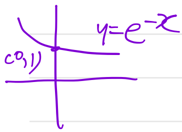
현재 해 θ의 이웃 중에 하나를 임의로 선정하여 넘겨줌 (현재해 변형)

최소화 문제에서는 J(θ)가 작을수록 큰 값을, 최대화 문제에서는 J(θ)가 클수록 큰 값을 넘겨준다.

→ 그렇지 않아도 여지 허용

변형해가 더 우수하면 그걸 취한다.

new가 더 큰 값이면 믿으로..



T클수록 많이 허용

두 해의 품질 차이가 작을수록 열등한 해가 채택되기 쉽도록

11.4 유전 알고리즘

$$\theta_{t+1} = \theta_t + N(0, \sigma)$$ 연산 추가

현재 해에 정규 분포로 얻은 임의 값을 합하는 것으로서 유전 알고리즘의 변이에 해당. 이 연산은 지역점에서 탈출하도록 도와줌. 진화 전략(evolution strategy)라 부름

11.4.1 원리 : 내리막 경사법, 시뮬레이티드 어닐링, 유전알고리즘

- 내리막 경사법, 시뮬레이티드 어닐링, 유전 알고리즘의 비유
  - 히말라야 산맥에서 가장 높은 에베레스트 봉을 찾는 문제

"오르막 경사법에서는 캥거루 한 마리가 가장 가파른 경사면만 따라 계속 산을 올라 꼭대기에 도달한다. 처음 시작한 곳이 에베레스트 봉에 속한 곳이 아니라면 절대 에베레스트 꼭대기에 도달하지 못한다. 시뮬레이티드 어닐링에서는 캥거루가 술에 취해 이리 저리 날 뛰다가 시간이 지남에 따라 술이 조금씩 깨어 가파른 경사면을 따라 올라 꼭대기에 도착한다. 시작점이 에베레스트가 아니더라도 날 뛰다가 우연히 에베레스트의 어느 곳에 도달하여 결국 에베레스트 꼭대기에 도착할 수도 있다. 유전 알고리즘에서는 비행기가 등장한다. 히말라야 산맥 곳곳에 여러 마리의 캥거루를 떨어뜨린다. 이들은 기를 쓰고 꼭대기로 올라가지 않는다. 그냥 있는 곳에서 오르락 내리락 거리며 풀을 뜯어 먹고 자식을 낳으며 산다. 몇 년에 한번씩 포수가 나타나 낮은 곳에 있는 캥거루를 잡아간다. 따라서 캥거루들이 사는 곳은 세대를 거듭하면서 높아진다."

알고리즘 [11.7] 유전 알고리즘

입력: 목적 함수 $J(\boldsymbol{\theta})$

출력: 최적해 $\hat{\boldsymbol{\theta}}$

알고리즘:

Population: 해의 집합

1. 초기 해 집단 P를 설정한다. // P는 여러 개의 해를 가짐
2. P의 해들의 품질을 평가한다. 어떻게?
3. P에서 가장 우수한 해를 $\hat{\boldsymbol{\theta}}$에 저장한다.
4. **repeat** {
5. P의 해들에게 적합도를 부여한다. 어떻게?
6. **for** ($i = 1$ **to** k) { 어떻게 자식해 k개 만들어줌
7. P에서 두 개의 해를 선택하여 이들을 $\boldsymbol{\theta}_1$과 $\boldsymbol{\theta}_2$라 한다.
8. $\boldsymbol{\theta}_i' = \text{crossover}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$; // 부모 해로부터 교차와 변이를 통해
9. mutation($\boldsymbol{\theta}_i'$); // 자식 해를 만든다.
10. $\boldsymbol{\theta}_i'$의 품질을 평가한다.
11. }
12. $\boldsymbol{\theta}_1', \cdots, \boldsymbol{\theta}_k'$를 P에 대치한다.

기준의 어떤 θ를 빼고?

13. $\boldsymbol{\theta}_1', \cdots, \boldsymbol{\theta}_k'$에서 가장 우수한 해가 $\hat{\boldsymbol{\theta}}$보다 좋으면 그것을 $\hat{\boldsymbol{\theta}}$로 한다.
14. } **until** (멈춤 조건);
15. **return** $\hat{\boldsymbol{\theta}}$;

11.4.2 유전 알고리즘의 구조

• 원리

- 라인 7의 선택: 적합도가 높은 해일 수록 선택될 확률이 높다. 하지만 못난 해도 0 이상의 확률을 갖는다.
- 라인 8의 교차로 만들어진 자식해는 부모해의 형질을 이어받음
- 라인 9의 돌연변이는 지역 최적점을 벗어날 가능성을 제공 (premature convergence 방지 역할)
- 자식해의 품질은 해 집단의 평균 품질을 넘을 가능성이 높다. 세대를 거듭할 수록 해 집단의 품질은 점점 좋아진다.

• 안정 상태형(steady-state)과 세대형(generational)
- 안정 상태형 : 라인 6-11에서 k=1로 두어 한 세대에 하나의 해 생산
- 세대형: k=n으로 두어 한 세대에 해집단 전체를 대치

• 해 표현과 초기 해 생성
- 문제에 적합한 해 표현을 개발해야 한다.
ex) 특징선택 : 이진 열, 신경망 훈련: 실수 코딩
- 초기해는 보통 난수를 이용하여 생성

• 멈춤 조건
- 수렴이 되면 멈춘다. 그런데 수렴 여부는 어떻게 알 수 있나?
- 주어진 문제에 적합하게, 보통 아래 조건을 적절히 and 또는 or 하여 사용
  · 주어진 계산 시간이 지났다.
  · 세대가 주어진 최대 세대에 도달하였다.
  · 여러 세대에 걸쳐 더 이상 좋은 해가 발생하지 X
  · 해 집단의 평균 적합도가 여러 세대에 걸쳐 더 이상 향상X

11.4.3 유전 연산

- 적합도 계산

① 해 품질을 그대로 사용

- 적절할 수도 그렇지 않을 수도 있음
- 초기에는 우열이 두드러지지만 세대가 지남에 따라 품질 값의 차이가 미세해져 더 이상 진화가 안일어 날 수도 있음

② 품질 비례 방법

- f_i는 i번째 해의 적합도. q_i는 i번째 해의 품질

좋은해일수록 ⇑

$$f_i = \underline{(q_i - q_{worst})} + (q_{best} - q_{worst})/(r-1)$$

- q_{best} 와 q_{worst}는 각각 가장 좋은 해와 가장 나쁜 해의 품질
- 가장 좋은 해는 가장 나쁜 해의 r배 적합도 (r이 클수록 선택 압력 높다.)

좋은 해의 선택 확률을 높이면 선택 압력이 높아진다.

③ 순위 기반 방법

- 해를 품질에 따라 정렬하고 순위에 따라 적합도를 부여

$$f_i = \underline{q}(1-q)^{i-1}$$

q 클수록 선택 압력 높다.

→ 때에 따라서 적합도 합이 1이 되도록 정규화할 필요가 있다. 이를 위해 각각의 적합도를 전체 적합도의 합으로 나누어 주면 된다.

예제 11.5 적합도 계산

해 집단의 크기가 $n = 4$라 하고 네 개의 해가 아래와 같은 품질을 가진다고 가정하자.

해1 = 32, 해2 = 69, 해3 = 63, 해4 = 72

해의 품질을 그대로 적합도로 취하는 첫 번째 방법에서는 적합도 f_i와 정규화 적합도 nf_i가 아래와 같다.

$f_1 = 32,\ f_2 = 69,\ f_3 = 63,\ f_4 = 72$
$nf_1 = 0.1356,\ nf_2 = 0.2924,\ nf_3 = 0.2669,\ nf_4 = 0.3051$

품질 비례 방법 ($r = 5$)과 순위 기반 방법 ($q = 0.3$)에 의한 적합도와 정규화 적합도는 아래와 같다.

품질 비례 방법: $f_1 = 10,\ f_2 = 47,\ f_3 = 41,\ f_4 = 50$
$nf_1 = 0.0676,\ nf_2 = 0.3176,\ nf_3 = 0.2770,\ nf_4 = 0.3378$

순위 기반 방법: $f_1 = 0.1029,\ f_2 = 0.21,\ f_3 = 0.147,\ f_4 = 0.30$
$nf_1 = 0.1354,\ nf_2 = 0.2764,\ nf_3 = 0.1934,\ nf_4 = 0.3948$ ■■■

- 공유 (sharing)

- 해 집단의 다양성을 높이려는 목적 (다양성이 높다는 것은 탐색 공간을 골고루 살펴본다는 뜻이므로 전역 최적 점에 도달할 가능성을 높여줌)

- 원리
→ 식 (11.13)으로 서로 비슷한 해의 적합도를 낮추어 줌

$$\hat{f}_i = \frac{f_i}{\sum_{j=1} s(d_{ij})}$$

* s(d)는 공유함수로 d가 클수록 작아지는 함수

→ d_{ij}는 해 i와 j 사이의 거리
→ s(d)는 d가 클수록 작아지는 함수

- 선택(selection)

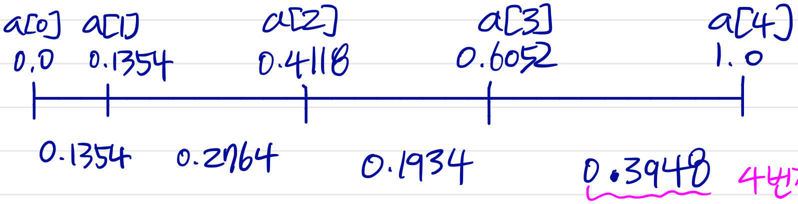
우수한 해가 선택될 확률이 커야 한다. 그리고 열등한 해도 낮은 확률이지만 선택될 기회가 주어져야 한다.

* 룰렛 휠 방법

r = random(); // [0, 1] 사이의 난수

$a[i-1] < r \leq a[i]$인 해 i를 선택하라.

예) $nf_1 = 0.1354$, $nf_2 = 0.2764$, $nf_3 = 0.1934$, $nf_4 = 0.3948$



4번째 해가 선택될 확률

그림 11.14 룰렛 휠 방법

* 토너먼트 방법:

[1, n] 사이의 서로 다른 두 개의 정수 i_1과 i_2를 임의로 생성하고 해 i_1과 i_2를 선택한다.

r = random(); // [0, 1] 사이의 난수

if(r < t) i_1과 i_2 중에 우수한 것을 선택하고,

else 열등한 것을 선택한다.

• 교차 (부모해로부터 자식 해 생성, 부모 형질 이어받음)

- 자름선 교차 (이진 염색체)

부모1 : 0 1 1 0 | 0 0 0 1 0 1 1 1

부모2 : 0 0 1 1 | 1 0 0 0 1 1 0 0

자식 : 0 1 1 0 1 0 0 0 1 1 0 0

부모1 : 0 1 | 1 0 0 | 0 0 1 0 | 1 1 1

부모2 : 0 0 | 1 1 1 | 0 0 0 1 | 1 0 0

자식: 0 1 1 1 1 0 0 1 0 1 0 0

그림 11.15 이진 열로 표현된 염색체의 교차 연산

• 교차(계속)

- 균등 교차(이진 염색체)

부모 1:	0	1	1	0	0	0	0	1	0	1	1	1	
부모 2:	0	0	1	1	1	0	0	0	1	1	0	0	
난수 :	0.3	0.8	0.4	0.9	0.4	0.6	0.5	0.7	0.4	0.3	0.1	0.9	($t = 0.5$)
자식 :	0	0	1	1	0	0	0	0	0	1	1	0	

그림 11.16 균등 교차

- 산술 교차(실수 염색체)

부모 1:	0.1	−0.5	0.3	0.7	−0.8
부모 2:	0.3	−0.3	0.5	0.9	0.6
자식 :	0.2	−0.4	0.4	0.8	−0.1

그림 11.17 산술 교차

• 변이 (부모해가 가지지 못한 형질을 부여하여 더 광범위한 공간 탐색)

- 이진 염색체의 변이 (변이 확률 P_m과 선택 압력의 관계는?)

변이전 :	0	0	1	1	0	0	0	0	0	0	1	0
난수 :	0.30	0.89	0.45	0.01	0.42	0.63	0.54	0.70	0.48	0.11	0.18	0.93
변이후 :	0	0	1	0	0	0	0	0	0	0	1	0

그림 11.18 변이 (p_m=0.02)

- 실수 염색체의 변이

변이 전 :	0.2	−0.4	0.4	0.8	−0.1	
난수 :	0.876	0.653	0.103	0.004	0.719	(p_m=0.005)
난수 :				−0.1		
변이 후 :	0.2	-0.4	0.4	0.7	−0.1	

그림 11.19 실수 염색체의 변이

난수 r의 범위는 문제의 성질과 유전자가 가지는 값의 범위에 따라 적절히 설정해야 함.

• 대치

알고리즘 [11.7]의 라인 6 에은 자식해를 K개 만들어 준다. 이제 이들을 해 집단에 넣어야 한다. 해 집단의 크기는 일정하므로 기존 해 K개를 들어 내고 자식 해를 넣어야 한다. 이를 대치라 한다.

11.4.4 매개 변수 설정과 선택 압력

- selection pressure
  - 선택 압력이 높다는 말은 우수한 해에게 기회를 더 줌을 뜻함
  - 너무 높으면 조기 수렴의 위험. 너무 낮으면 수렴 시간이 너무 김
  - 다양성과 선택압력은 반비례 관계이다.

- 탐사와 탐험
  - 탐사는 가능성 높아 보이는 지점을 집중적으로 찾아보는 성향
  - 탐험은 탐색 공간 전체를 고루고루 찾아보는 성향

- 유전 알고리즘이 추구하는 바는
  - 탐사와 탐험의 적절한 조화
  - 그러기 위해서는 여러 매개 변수를 조화롭게 설정해야 함

표 11.2 유전 알고리즘의 매개 변수와 선택 압력

연산 종류	매개 변수	선택 압력
해 집단	해의 개수 n이 클수록	낮음
순위 기반 적합도 계산	q가 클수록	높음
품질 비례 적합도 계산	r이 클수록	높음
토너먼트 선택	t가 클수록	높음
교차	자름 선이 많을수록	낮음
변이	p_m이 클수록	낮음
공유 ?	사용하면	낮음
엘리티즘	사용하면	높음
대치	가장 나쁜 해를 대치하면	높음

가장 좋은 해5 다음세대에도 보장

11.4.5 찬반 논쟁

찬
- 새로운 개념과 연산들이 재미있다.
- 기존 알고리즘의 한계를 극복할 것 같다.
- 시간을 더 주면 더 좋은 해를 기대할 수 있다.
- 문제에 대한 제약 조건이 적고 응용 범위가 넓다.
- 실제 실험해 보니 성능이 기존 알고리즘보다 좋다.

반
- 수학적 토대가 약하여 미덥지 못하다.
- 수행할 때마다 다른 해를 내면 어떡하나?
- 수행 시간이 길어 실시간 환경에서는 활용이 불가능하다.
- 실제 실험해 보니 성능이 기존 알고리즘보다 나쁘다.