韓國數學敎育學會誌 시리즈 E <數學敎育 論文集>
https://doi.org/10.7468/jksmee.2023.37.1.65
제 37집 제 1호, 2023. 3. 65-84

J. Korea Soc. Math. Ed. Ser. E:
Communications of Mathematical Education
Vol. 37, No. 1, Mar. 2023. 65-84

라그랑주 승수법의 교수·학습에 대한 소고: 라그랑주 승수법을 활용한 주성분 분석 사례

이 상 구 (성균관대학교, 교수)
남 윤 (성균관대학교, 연구원)
이 재 화 (성균관대학교, 연구원)†

라그랑주 승수법(Method of Lagrange Multipliers)은 등식 제약조건하에서 미분가능한 함수의 최대, 최소를 구하는 대표적인 방법이다. 선형대수학, 최적화 이론, 제어 이론을 포함하여 최근에는 인공지능 기초수학에서도 널리 활용되고 있다. 특히 라그랑주 승수법은 미분적분학과 선형대수학을 연결하는 중요한 도구이며, 주성분 분석(Principal Component Analysis, PCA)을 포함한 인공지능 알고리즘에 많이 활용되고 있다. 따라서 교수자는 대학 미분적분학에서 처음 라그랑주 승수법을 접하는 학생들에게 구체적인 학습 동기를 제공할 필요가 생겼다.
이에 본 논문에서는 교수자가 학생들에게 라그랑주 승수법을 효과적으로 교육하는데 필요한 통합적인 시야를 제공한다. 먼저 다양한 전공의 학생들이 계산에 대한 부담을 덜고 원리를 쉽게 이해할 수 있도록 개발한 시각화 자료 및 파이썬(Python) 기반의 SageMath 코드를 제공한다. 또한 라그랑주 승수법으로 행렬의 고윳값과 고유벡터를 유도하는 과정을 상세히 소개한다. 그리고 라그랑주 승수법을 간단한 경우에 대한 증명에서 시작하여 일반화된 최적화 문제로 확장하고, 수업에서 학생들이 라그랑주 승수와 PCA를 활용하여 실제 데이터를 분석한 결과를 추가하였다. 본 연구는 대학수학을 지도하는 다양한 전공의 교수자들에게 도움이 될 기초자료가 될 것이다.

Ⅰ. 서론

라그랑주 승수법(Method of Lagrange Multipliers)은 최적화 문제, 특히 등식 제약조건하에서 미분가능한 함수의 최대, 최소를 구하는 문제를 해결하는 기본적인 방법으로, 미분적분학을 비롯하여 선형대수학, 최적화 이론, 제어 이론(Control Theory) 등 수학의 다양한 분야에서 활용되고 있다. 또한 자연과학계열, 경상계열, 공학계열 등 다양한 전공에서도 최적화 문제를 다루고 있으며, 최근 유망한 분야인 기계학습(Machine Learning), 인공지능(Artificial Intelligence)에서도 최적화 문제를 해결하기 위해 라그랑주 승수법을 활용하고 있으므로, 다양한 전공의 학생들에게 라그랑주 승수법을 효과적으로 교육하는 것이 필요한 시점이다.

최적화(optimization)는 우리의 일상생활 속 의사결정에서도 자주 발생한다. 예를 들어, 집에서 학교나 직장까지 출퇴근할 가장 짧은 경로를 찾는다든가, 여행을 떠나기 전 시간과 비용의 제약을 고려해 가장 가고 싶은 곳을 가려는 계획을 짤 때 그러하다. 자연과학, 공학, 사회과학 등에서 발생하는 문제들도 마찬가지이다. 그래서 미분적분학에서의 최대-최소문제, 선형대수학에서의 고윳값(eigenvalue) 문제, 수치해석학에서의 다양한 문제들이

* 접수일(2022년 12월 2일), 심사(수정)일(2022년 12월 25일), 게재 확정일(2022년 12월 28일)
* MSC2000 분류 : 97U50, 97U60, 97U70
* 주제어 : 라그랑주 승수법, 미분적분학, 선형대수학, 최적화, 제어이론, 주성분 분석, 이차형식, KKT 조건
† 교신저자 : jhlee2chn@skku.edu
* 이 논문은 2020년도 정부(과기정통부)의 재원으로 한국연구재단 혁신성장 선도 고급연구인재 육성사업(No.2020M3H1A1077095)과 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2021R1F1A1046714).

"maximize (minimize) $f(x, y)$ subject to $g(x, y) \geq 0,\ h(x, y) = 0$"

와 같이 '변수가 특정한 제약조건(주로 방정식이나 부등식으로 나타난다)을 만족할 때, 어떤 함수를 최대화하거나 최소화하는 형태의 최적화 문제'로 나타난다.

라그랑주 승수법은 함수의 최대, 최소를 구하는 문제와 관련하여 대학 미분적분학(다변수 미분적분학)에서 지도되고 있다. 그러나 주로 함수의 최대, 최소를 어떻게 구하는지에 관한 방법적인 측면만 강조하고, 학생들이 라그랑주 승수법을 왜 배우는지, 라그랑주 승수가 무엇을 의미하는지에 대한 이해를 간과하는 경우가 많으며, 문제가 조금만 복잡해져도 실제 지필로는 계산하기 힘든 경우가 많다. 특히 다양한 전공의 학생들이 미분적분학과 함께 많이 수강하는 선형대수학에서 주요 개념인 대칭행렬 A의 고윳값이 제약조건 $\mathbf{x}^T\mathbf{x} = 1$하에서 이차형식(quadratic form)[1] $f(\mathbf{x}) = \langle A\mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^T A\mathbf{x}$의 최대, 최소를 구하는 문제에서의 라그랑주 승수라는 사실이 제대로 전달되지 못하고 있다. 그래서 라그랑주의 승수가 미분적분학과 선형대수학을 잇는 중요한 개념이라는 것을 이해하지 못하고, 각 과목에서 별개의 내용으로 인식하여 일부 측면만 이해하게 된다. 더불어, 인공지능에서 데이터의 차원을 축소하는 대표적인 알고리즘인 주성분 분석(Principal Component Analysis, PCA)은 결국 공분산 행렬의 고윳값과 고유벡터를 구하는 과정인데, 라그랑주 승수법을 이해하지 못하게 되면, PCA의 이와 같은 본질을 제대로 파악할 수 없게 된다. 따라서 인공지능 기초수학을 학습하는 대학 신입생들은 PCA를 학습하기 전에 미분적분학에서 라그랑주 승수법을 디지털 시대에 맞게 학습하여야 한다.

이에 본 논문에서는 대학 미분적분학 교수자가 학생들에게 라그랑주 승수법을 효과적으로 교육할 수 있도록 필요한 지식을 종합하여 제시한다. 먼저 교수자들이 다양한 전공의 학생들에게 라그랑주 승수법을 이해시킬 수 있도록 통합적인 설명을 시각화 자료와 함께 제시하고, 복잡한 문제도 어려움 없이 계산할 수 있도록 파이썬(Python) 기반의 SageMath 코드도 제공한다. 이를 활용하면 제시된 코드에서 함수와 조건들을 변경해가면서 유사한 문제도 쉽게 해결할 수 있다. 또한 행렬의 고윳값 문제를 라그랑주 승수법으로 유도하는 과정을 자세히 제공하여, 선형대수학과의 연결성을 충분히 이해하도록 한다. 그리고 라그랑주 승수법이 처음 활용되는 최적화 문제 중 1계 최적 조건(First-Order Optimality Condition)의 증명에서 시작하여, 일반적인 최적화 문제의 경우로 발전시킨 후, 라그랑주 승수법를 실제 데이터에 적용한 예를 소개한다.

Ⅱ. 연구의 배경

1. 배경지식

조제프 루이 라그랑주(Joseph-Louis Lagrange, 1736-1813)는 1788년 Méchanique Analitique라는 책에서 라그랑주 승수법을 현재 알려진 바와 같은 형식으로 제시하였다. 그는 정역학(statics)의 관점에서 특정한 조건들을 따르는 '물질 점들의 계(system of material points)'가 이루는 '평형상태(equilibrium)'를 분석하는 과정이, 수학적으로는 제약조건하에서 어떤 함수의 최대 또는 최소를 찾는 문제의 최적해가 되기 위한 필요조건을 결정하는 것과 같음을 보였다. 이때 새로운 변수인 승수(multiplier)를 도입하여 기존의 제약조건을 평형상태를 나타내는 방정식으로 일관되게 표현하였다. 이 새로운 방법은 라그랑주의 이름을 따서 명명되었지만 실제로는 의사이자 수학 교수였던 웨어링(Edward Waring, ca. 1736 - 1798)이 1779년 처음 발견했다고 한다. 그리고 1783년 오일러(Leonhard Euler, 1707-1783)에 의해 간단한 결과가 공개되기도 하였다. 한편 라그랑주는 앞의 책에서 '강체(rigid body)'의 평형상태의 분석을 위해 '변분법(Calculus of Variation)'으로 알려진 수학적 방법도 고안하였다

1) 이차형식(quadratic form) http://matrix.skku.ac.kr/2014-Album/Quadratic-form/

(Bussotti, 2003). 변분법은 제약조건을 갖는 ‘범함수(functional)’의 최대, 최소를 찾는 방법으로 라그랑주 승수법을 일반화한 것이다.

라그랑주 승수법은 <표 Ⅱ-1>과 같이 등식 제약조건하에서 미분가능한 함수의 최대, 최소를 구하는 문제를 해결하는 기본적인 방법이다.

<표 Ⅱ-1> 등식 제약조건이 있는 최적화 문제(제약조건 m개, 독립변수 n개)

$$\begin{array}{lll} \text{minimize (maximize)} & f(\mathbf{x})=f(x_1,x_2,\dots,x_n) & \mathbf{x}=(x_1,x_2,\dots,x_n)\in\mathbb{R}^n \\ \text{subject to} & c_1(\mathbf{x})=c_1(x_1,x_2,\dots,x_n)=0 & \\ & c_2(\mathbf{x})=c_2(x_1,x_2,\dots,x_n)=0 & \\ & \cdots & \\ & c_m(\mathbf{x})=c_m(x_1,x_2,\dots,x_n)=0 & \end{array}$$

이때 f를 목적함수(objective function)라 하고, $c_i(\mathbf{x})=0,\ i=1,\dots,m$을 등식 제약조건(equality constraint)이라 한다. 그러면 $\mathbf{x}^*=(x_1^*,x_2^*,\dots,x_n^*)$가 <표 Ⅱ-1> 문제의 최적해가 되기 위한 1계 최적 조건은 다음과 같다.

정리 1. 점 $\mathbf{x}^*$가 <표 Ⅱ-1> 문제의 최적해라 하자. 만일 $\mathbf{x}^*$에서 $\nabla c_i(\mathbf{x}^*),\ i=1,\dots,m$가 일차독립이면, 적당한 상수 $\lambda_i^*,\ i=1,\dots,m$가 존재하여 다음을 만족한다.

$$\nabla f(\mathbf{x}^*)=\sum_{i=1}^{m}\lambda_i^*\nabla c_i(\mathbf{x}^*),\quad c_i(\mathbf{x}^*)=0,\ i=1,\dots,m$$

이때 λ_i^*를 라그랑주 승수라 한다.

즉 라그랑주 승수법은 정리 1의 결과를 바탕으로 다음과 같이 최적해가 될 후보들을 계산하는 방법이다.

<표 Ⅱ-2> 라그랑주 승수법

[단계 1] 다음 연립방정식을 만족하는 $\mathbf{x}$, λ_i $(i=1,\dots,m)$의 값을 모두 구한다.

$$\nabla f(\mathbf{x})=\sum_{i=1}^{m}\lambda_i\nabla c_i(\mathbf{x}),\quad c_i(\mathbf{x})=0,\ i=1,\dots,m$$

[단계 2] [단계 1]에서 구한 모든 점 $\mathbf{x}$에서 f의 값을 계산한다. 이 중에서 가장 큰 값이 f의 최댓값이고, 가장 작은 값이 f의 최솟값이다.

라그랑주 승수법은 다음과 같이 라그랑주 함수(Lagrange function)

$$L(\mathbf{x},\boldsymbol{\lambda})=L(x_1,\dots,x_n,\lambda_1,\dots,\lambda_m)=f(x_1,\dots,x_n)-\sum_{i=1}^{m}\lambda_i c_i(x_1,\dots,x_n)$$

를 도입하여 설명할 수도 있다(Chiang & Wainwright, 2004). 즉 라그랑주 승수법의 [단계 1]은 $\nabla L(\mathbf{x},\boldsymbol{\lambda})=\mathbf{0}$을 만족하는 $\mathbf{x}$, $\boldsymbol{\lambda}$를 찾는 것과 같다. 그래서 라그랑주 승수법은 제약조건이 있는 최적화 문제를 제약조건이 없는 최적화 문제로 변형시켜 적용할 수 있는 방법으로 설명되기도 한다. 라그랑주 함수는 특히 독립변수와 제약조건의 개수가 많고 복잡한 일반적인 최적화 문제로 라그랑주 승수법을 확대, 적용할 때 유용하다.

2. 라그랑주 승수법 교수·학습에서의 한계점

라그랑주 승수법을 설명하기 위해서는 다변수 함수의 그래디언트 개념이 필요하므로, 주로 대학 미분적분학에서 처음 다루어진다. 그러나 함수의 최대, 최소를 어떻게 구하는지에 관한 방법적인 측면(<표 Ⅱ-2>)이 강조되어왔기 때문에 문제가 조금만 복잡해지면 지필로 계산하기 어려워져서 실제 강의에서는 아주 단순한 형태의 문제만 주로 다루게 된다. 본 연구에서는 이를 극복하는 방법으로 선형대수학적 지식과 코딩을 이용하는 일반적인 해법을 소개한다. 먼저 기존의 교재(Lee et al., 2014)에서 제시한 다음 예제를 보자.

예제 1. 타원면(ellipsoid) $x^2+4y^2+3z^2=12$에서 점 $(2,\ -1,\ 2)$까지의 최단거리와 최장거리를 구하여라.

그러면 점 $(2,-1,2)$에서 $\mathbb{R}^3$의 임의의 점 (x,y,z)까지의 거리공식 $\sqrt{(x-2)^2+(y+1)^2+(z-2)^2}$ 으로부터 이 문제는 방정식 $x^2+4y^2+3z^2=12$를 만족하면서 $d=\sqrt{(x-2)^2+(y+1)^2+(z-2)^2}$ 가 최대, 최소가 되는 점 (x,y,z)를 찾는 문제가 됨을 알 수 있다. 따라서 수학적으로는 다음과 같이 나타낼 수 있다.

minimize (maximize)	$f(x,y,z)=d^2=(x-2)^2+(y+1)^2+(z-2)^2$	$(x,y,z)\in\mathbb{R}^3$
subject to	$g(x,y,z)=x^2+4y^2+3z^2-12=0$	

이제 라그랑주 승수법을 이용하면 [단계 1]에서 다음과 같은 연립방정식을 얻는다.

$$2x-4=2\lambda x,\ 2y+2=8\lambda y,\ 2z-4=6\lambda z,\ x^2+4y^2+3z^2=12.$$

첫 번째, 두 번째, 세 번째 식으로부터 $x=\dfrac{2}{1-\lambda}$, $y=-\dfrac{1}{1-4\lambda}$, $z=\dfrac{2}{1-3\lambda}$임을 알 수 있고, 이를 네 번째 식에 대입하면 $\left(\dfrac{2}{1-\lambda}\right)^2+4\left(-\dfrac{1}{1-4\lambda}\right)^2+3\left(\dfrac{2}{1-3\lambda}\right)^2=12$라는 식을 얻는다. 그러나 이 식을 만족하는 λ를 지필로 계산하여 구하는 것은 쉽지 않다. 즉 코딩/컴퓨터를 활용하지 않는다면 대부분의 현실 문제에 대한 해를 구하는 것이 불가능하다. 그러나 <표 Ⅱ-3>과 같이 코드를 사용하면 연립방정식의 해들을 쉽게 구할 수 있으며, 유사한 문제도 함수와 조건을 변경해가면서 모두 해결할 수 있다.

이 예제에서 구한 λ는 $\lambda\fallingdotseq-0.1091$와 $\lambda\fallingdotseq1.602$이다. 먼저 $\lambda\fallingdotseq-0.1091$일 때, 해는 $(1.803,\ -0.6961,\ 1.507)$이고 함숫값은 $f(x,y,z)=0.3745$이다. 그리고 $\lambda\fallingdotseq1.602$일 때, 해는 $(-3.322,\ 0.1849,\ -0.5255)$이고 함숫값은 $f(x,y,z)=36.10$이다. 두 함숫값을 비교하면 최단거리는 약 0.6119이고 최장거리는 6.009임을 알 수 있다. 이와 같이 한 점과 타원면 사이의 최단거리와 최장거리를 찾는 비교적 단순한 문제도 상황에 따라서는 손으로 해결하기 힘든 복잡한 계산이 필요할 수 있다.

<표 II-3> 예제 1을 해결하는 라그랑주 승수법 코드(SageMath)

```
var('x, y, z, ld')
f(x, y, z) = (x - 2)^2 + (y + 1)^2 + (z - 2)^2
g(x, y, z) = x^2 + 4*y^2 + 3*z^2 - 12
gradf = f.gradient()
gradg = g.gradient()
cpoints = solve([gradf[0] == ld*gradg[0], gradf[1] == ld*gradg[1], gradf[2] == ld*gradg[2], g(x, y, z) == 0], x, y, z, ld,
solution_dict = True)
for sol in cpoints:
    if ((sol[x] in RR) and (sol[y] in RR) and (sol[z] in RR) and (sol[ld] in RR)):
        print("x =", sol[x].n(digits = 4), ",", "y =", sol[y].n(digits = 4), ",", "z =", sol[z].n(digits = 4), ",", "f(x, y, z) =",
f(x = sol[x], y = sol[y], z = sol[z]).n(digits = 4), ",", "lambda = ", sol[ld].n(digits = 4))
```

또한 라그랑주 승수법은 선형대수학과 미분적분학을 연결하는데 사용된다. 예를 들어, 선형대수학 교재에서는 $A\mathbf{x} = \lambda\mathbf{x}$를 만족하는 스칼라 λ와 $\mathbf{0}$이 아닌 벡터 $\mathbf{x}$를 각각 A의 고윳값과 고유벡터로 정의한 후, 여기서 구한 고윳값의 의미에 대하여 설명을 하는 대신 고윳값을 구하는 데에만 집중하고 있다. 그러나 대칭행렬 A의 고윳값과 고유벡터를 계산하는 문제가, 제약조건 $\mathbf{x}^T\mathbf{x} = 1$하에서 이차형식 $f(\mathbf{x}) = \langle A\mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^T A\mathbf{x}$의 최대, 최소를 구하기 위해 미분적분학에서 학습한 라그랑주 승수법을 적용하여 라그랑주 승수를 계산하는 문제와 일치한다는 것을 설명하는 교재는 거의 없다. 마찬가지로 대부분의 다변수 미분적분학 교재에서도 행렬의 고윳값과 고유벡터를 구하는 데 라그랑주 승수법이 활용된다는 사실을 언급하지 않고 있다. 따라서 이 두 가지 방식을 서로 다른 개념으로 생각하면 학생들이 학습할 때 노력을 배로 들이게 되고, 그 다음 단계의 학습에 지장을 받게 된다.

3. 연구방법 및 절차

본 연구진은 앞서 언급한 문제를 해결하기 위해 라그랑주 승수법을 다양한 전공의 학생에게 쉽고 효과적으로 교육할 수 있는 교수학습자료가 필요하다고 판단하였다. 그래서 라그랑주 승수법과 관련된 논문과 교재 등을 충분히 검토한 후, 대학 미분적분학 교수자의 입장에서 본문에 제시한 자료만으로도 라그랑주 승수법을 교육하기에 충분하도록 구성하고자 하였다. 이와 같은 방식은 서보억(2021)의 연구에서 제시한 수학교육적 탐구방법 가운데 한 가지 사례로 볼 수 있다. 본 연구진은 다음과 같은 방법으로 자료를 정리하였다.

① 먼저 다양한 전공의 학생들이 라그랑주 승수법을 이해할 수 있도록, 쉬운 개념과 간단한 예제로 시각화 자료를 구성하여 설명하였다.

② 학생들이 수학에 관해 생각하고 논의할 시간을 충분히 갖도록 복잡한 계산에 대한 부담을 덜어주었다. 특히 파이썬 기반의 SageMath 코드를 제시하여 함수와 조건들을 변경해가면서 유사한 문제도 쉽게 해결할 수 있도록 하였다.

③ 특히 선형대수학에서 다루는 행렬의 고윳값 문제를 라그랑주 승수법으로 유도하는 과정을 자세히 제공하여, 선형대수학과의 연결성을 충분히 이해하도록 하였다. 이는 특히 인공지능에서 PCA의 본질을 쉽게 파악할 수 있도록 한다.

④ 심화된 내용을 학습하기 원하는 학생들을 위해 최적화 문제의 1계 최적 조건에 대한 증명과 함께 라그랑주 승수의 의미도 쉽게 설명하였다.

⑤ 1계 최적조건을 부등호 제약조건을 갖는 더 일반적인 최적화 문제로 확장한 '카루시-쿤-터커 조건(Karush-Kuhn-Tucker Condition, KKT 조건)'에 대해서도 간단히 언급하였다.

⑥ 라그랑주 승수법의 응용 사례로, PCA를 학생들이 학습하여 실제 데이터를 분석한 결과를 소개하였다.

Ⅲ. 연구 결과 및 논의

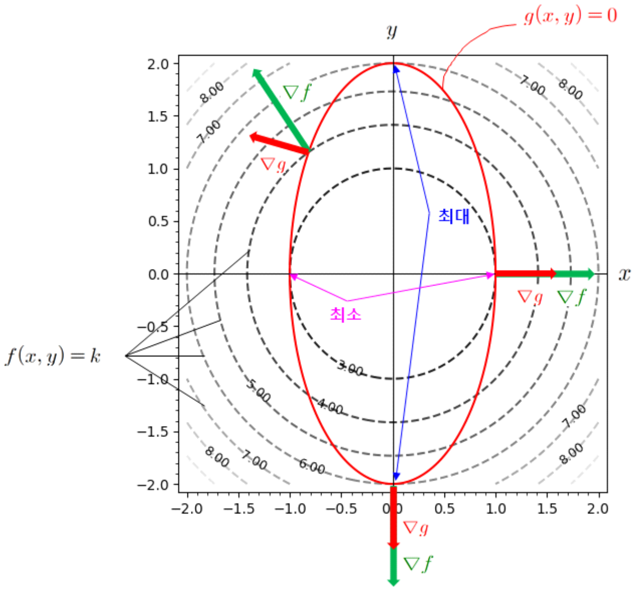
이 장에서는 라그랑주 승수법을 설명하는 시각화 자료와 SageMath 코드, 선형대수학과 주성분 분석에서의 활용과 실습사례 및 라그랑주 승수의 의미 등에 대하여 설명한다.

1. 시각화 자료 및 SageMath 코드

f가 2변수 또는 3변수 함수일 때는 등위곡선(level curve) 또는 등위곡면(level surface)을 그릴 수 있는데, 이 때 그래디언트는 주어진 점을 통과하는 등위곡선 또는 등위곡면에 수직이다. 그리고 n변수 함수 f의 경우 ∇f는 점 $\mathbf{x} = (x_1, x_2, \cdots, x_n)$에서의 f의 최대 증가율의 방향을 나타내며, 그 최대 변화율은 $\|\nabla f(\mathbf{x})\|$이다. 우리는 제약조건 $g(x, y) = x^2 + \frac{1}{4}y^2 - 1 = 0$을 만족하는 x, y에 대하여 함수 $f(x, y) = 2 + x^2 + y^2$의 최댓값과 최솟값을 구하는 문제(이상구 외, 2022)를 예로 들어, 다변수 미분적분학을 처음 학습하는 학생들이 라그랑주 승수법의 원리를 직관적으로 이해할 수 있도록 <표 Ⅲ-1>의 본문과 [그림 Ⅲ-1]과 같이 이미지를 생성하여 시각적으로 설명하였다.

<표 Ⅲ-1> 라그랑주 승수법에 관한 본문 설명

먼저 방정식 $g(x,y)=0$으로 주어지는 곡선(빨간색)과 적당한 실수 k값들에 대해 등위곡선(level curve) $f(x,y)=k$의 그래프(검은색, 회색)를 좌표평면에 같이 그려보자. 아래 그림을 통해 (x,y)가 방정식 $g(x,y)=0$으로 주어진 곡선(빨간색) 위를 움직이는 동안 $f(x,y)$가 최대, 최소가 되도록 하는 점(**최적해, optimal solution**)은 $g(x,y)=0$와 $f(x,y)=k$이 접할 때임을 쉽게 알 수 있다. 특히 등위곡선에 적혀있는 함숫값들을 통해 $f(x,y)$는 원점으로부터 바깥쪽으로 퍼져나가는 방향으로 증가하는 함수임을 알 수 있으므로, f가 최대(파란색)가 되는 점은 y축 상에 두 군데 나타나고, 최소(분홍색)가 되는 점은 x축 상에 두 군데 나타난다. 함수 $f(x,y)$에 대하여 f의 그래디언트(gradient) $\nabla f(x,y)$는 등위곡선 $f(x,y)=k$에 직교한다. 따라서 $g(x,y)=0$을 만족하는 x,y에 대하여 함수 $f(x,y)$를 최대, 최소가 되게 하는 점 (x,y)에서는 $g(x,y)=0$와 $f(x,y)=k$이 접하게 되므로 $\nabla f(x,y)$와 $\nabla g(x,y)$는 평행, 즉 적당한 실수 λ가 존재하여 $\nabla f(x,y)=\lambda \nabla g(x,y)$을 만족하게 된다. 이러한 λ를 **라그랑주의 승수(Lagrange multiplier)**라 한다. 그림의 왼쪽 상단의 점에서와 같이 만일 (x,y)에서 $\nabla f(x,y)$와 $\nabla g(x,y)$가 평행하지 않으면, (x,y)가 곡선 $g(x,y)=0$을 따라 움직일 때, 함수 f가 현재보다 증가할 여지가 있다. 실제로 이 점에서 시계 방향으로 $g(x,y)=0$을 따라 움직이면 함수 f의 값이 증가하게 된다. 즉 그래디언트와 등위곡선이 직교하는 점에서 2개의 그래디언트가 평행하고, 그 두 벡터 사이의 관계가 라그랑주 승수가 된다는 것이다.

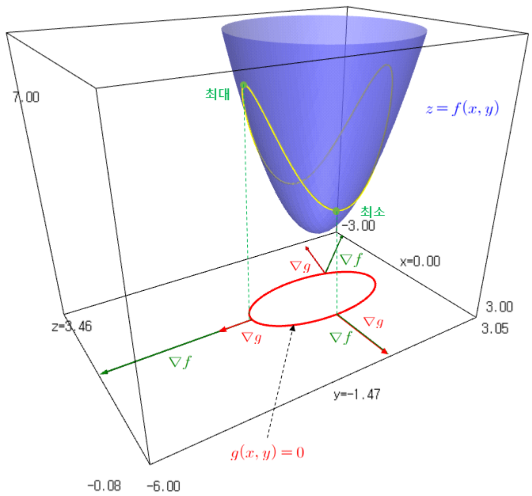


[그림 III-1] 라그랑주 승수법을 이해하기 위한 시각화 자료

이때 지필만으로는 목적함수의 등위곡선과 제약조건의 그래프를 그리기 어렵다. 아래와 같이 SageMath 코딩을 활용하면 손쉽게 그릴 수 있으며, 학습자는 실습실의 코딩을 간단히 조작하는 것으로 다양한 시도를 할 수 있다. 이러한 그래프를 활용하면, 라그랑주 승수법의 기본 원리를 쉽게 설명할 수 있다. 아래 명령어를 복사하여 웹 사이트 http://sage.skku.edu/에 입력 후 실행하면 위의 [그림 Ⅲ-1]을 얻을 수 있다.

```
var ('x, y')
f(x, y) = 2 + x^2 + y^2  # 목적함수 f(x, y)
g(x, y) = x^2 + 1/4*y^2 - 1  # 제약조건 g(x, y) == 0의 함수
p1 = contour_plot(f, (x, -2, 2), (y, -2, 2), fill = False, labels = True, linestyles = 'dashed', label_colors = 'black')
p2 = implicit_plot(g(x, y) == 0, (x, -2, 2), (y, -2, 2), color = 'red', axes = True)
p1 + p2
```

위의 내용을 3차원 공간상에 [그림 Ⅲ-2]와 같이 나타내면 좀 더 직관적으로 라그랑주 승수법의 원리를 이해할 수 있다. [그림 Ⅲ-2]는 본 연구진이 SageMath 코드를 이용하여 실제로 생성한 이미지로 주어진 목적함수가 어디에서 최댓값, 최솟값을 갖는지와 라그랑주 승수가 어떤 의미를 갖는지를 구체적으로 보여준다.



[그림 III-2] [그림 III-1]을 3차원상에 나타낸 그림

이제 라그랑주의 승수법으로 다음의 SageMath 코드를 활용하여 위의 예제를 풀면 점 $(1, 0)$, $(-1, 0)$, $(0, 2)$, $(0, -2)$를 얻는다.

```
var ('x, y, ld')
f(x, y) = 2 + x^2 + y^2
g(x, y) = x^2 + 1/4*y^2 - 1
gradf = f.gradient()
gradg = g.gradient()
solve ( [gradf[0] == ld*gradg[0], gradf[1] == ld*gradg[1], g(x, y) == 0], x, y, ld )
```

그리고 이 점들에서 f의 함숫값을 구하여 비교하면 최솟값은 $f(1, 0) = f(-1, 0) = 3$ (이때 $\lambda = 1$), 최댓값은 $f(0, 2) = f(0, -2) = 6$ (이때 $\lambda = 4$)이 된다.

라그랑주 승수법은 $n(\geq 3)$개의 독립변수를 갖는 다변수 함수와 $m(\geq 2)$개의 제약조건들이 있는 경우에도 적용할 수 있다. 대학 미분적분학 수업에서는 학생들의 수학적 기호를 다루는 훈련의 부족으로, 곧바로 일반화한 정리로 나아가기에는 무리가 있을 수 있다. 그래서 <표 Ⅲ-2, Ⅲ-3>과 같이 3변수 함수와 2개의 제약조건이 있는 경우를 예로 들어, 나중에 더 복잡한 문제에 대한 라그랑주 승수법도 큰 어려움 없이 접근할 수 있도록 한다.

<표 III-2> 3변수 함수와 2개의 제약조건을 갖는 최적화 문제에 대한 라그랑주 승수법

두 개의 방정식 $g(x, y, z) = 0$와 $h(x, y, z) = 0$를 만족하는 x, y, z에 대하여, 함수 $f(x, y, z)$의 최댓값과 최솟값을 구하는 문제는 라그랑주 승수법으로 다음과 같이 해결한다. 여기서 아래의 λ와 μ를 라그랑주 승수라 한다. [단계 1] 다음 연립방정식을 만족하는 x, y, z, λ, μ의 값을 모두 구한다. $\nabla f(x, y, z) = \lambda \nabla g(x, y, z) + \mu \nabla h(x, y, z)$ $g(x, y, z) = 0, \quad h(x, y, z) = 0$ [단계 2] [단계 1]에서 구한 모든 점 (x, y, z)에서 f의 값을 계산한다. 이 중에서 가장 큰 값이 f의 최댓값이고, 가장 작은 값이 f의 최솟값이다.

<표 III-3> 3변수 함수와 2개의 제약조건을 갖는 최적화 문제의 풀이 예시

예제 2. 평면 $2x+y+2z=9$과 $5x+5y+7z=29$에서 $f(x,\ y,\ z)=x^2+y^2+z^2$의 최솟값을 구하라.
풀이. $g(x,y,z)=2x+y+2z-9=0$와 $h(x,y,z)=5x+5y+7z-29=0$라 하고, 다음 연립방정식을 만족하는 $x,\ y,\ z,\ \lambda,\ \mu$의 값을 모두 구한다.

$$\nabla f(x,y,z)=\lambda\nabla g(x,y,z)+\mu\nabla h(x,y,z),\ g(x,y,z)=0,\quad h(x,y,z)=0$$

```
var ('x, y, z, l, m')
f(x, y, z) = x^2 + y^2 + z^2
g(x, y, z) = 2*x + y + 2*z - 9
h(x, y, z) = 5*x + 5*y + 7*z - 29
gradf = f.gradient(); gradg = g.gradient(); gradh = h.gradient()
solve ([gradf[0] == l*gradg[0] + m*gradh[0], gradf[1] == l*gradg[1] + m*gradh[1], gradf[2] == l*gradg[2] +
m*gradh[2], g(x, y, z) == 0, h(x, y, z) == 0], x, y, z, l, m)
```

```
[[x == 2, y == 1, z == 2, l == 2, m == 0]]
```

그러면 임계점 $(2,1,2)$를 얻는다. 이 점에서 함수 f의 극값을 구하면 다음과 같다.

```
var ('x, y, z, l, m')
f(x, y, z) = x^2 + y^2 + z^2
print ( f(2, 1, 2) )
```

```
9
```

여기서 3변수 함수와 2개의 제약조건을 갖는 문제에 대해 코딩을 활용한 풀이는 2변수 함수와 하나의 제약조건만 있는 문제의 풀이와 근본적으로 다르지 않다. 즉 4변수 함수의 경우도 위의 코드를 변경하여 같은 방식으로 해결할 수 있다. 따라서 변수의 개수가 많아 지필로 풀었을 때 발생할 수 있는 어려움을 피하면서도 충분히 라그랑주 승수법의 개념을 이해하고 활용할 수 있다. 이와 같이 코딩을 활용하면 개념을 일반화하는데 큰 어려움 없이 기존 코드에 약간의 변형과 첨가만으로도 문제를 해결하는 법을 익힐 수 있다.

2. 라그랑주 승수법과 선형대수학

이 절에서는 대칭인 정사각행렬의 고윳값 문제를 라그랑주 승수법으로 유도하는 과정(Horn & Johnson, 2013)을 제시한다.

$A=[a_{ij}]_{n\times n}$를 대칭인 정사각행렬이라 하고, 방정식 $\mathbf{x}^T\mathbf{x}=1$ $(g(\mathbf{x})=\mathbf{x}^T\mathbf{x}-1=0)$을 만족하는 벡터 $\mathbf{x}\in\mathbb{R}^n$에 대하여 함수 $f(\mathbf{x})=\langle A\mathbf{x},\mathbf{x}\rangle=\mathbf{x}^T(A\mathbf{x})=\mathbf{x}^TA\mathbf{x}$를 최대(또는 최소)화하는 문제를 생각해보자. 앞서 학습한 라그랑주 승수법을 적용하면, f를 최대 또는 최소가 되도록 하는 $\mathbf{x}$는 $\nabla f(\mathbf{x})=\lambda\nabla g(\mathbf{x})$을 만족해야 한다. 여기서 λ는 라그랑주 승수이다.

이제 $\nabla f(\mathbf{x})$와 $\nabla g(\mathbf{x})$가 어떻게 표현되는지 살펴보자. 먼저 $f(\mathbf{x})=\langle A\mathbf{x},\mathbf{x}\rangle=\mathbf{x}^TA\mathbf{x}=\sum_{i=1}^n\sum_{j=1}^n a_{ij}x_ix_j$에서 A가 대칭행렬이면 비대각선 성분이 $a_{ij}=a_{ji}$이므로 $i\neq j$일 때 $\mathbf{x}^TA\mathbf{x}$이 갖는 x_ix_j의 계수는 $a_{ij}+a_{ji}=2a_{ij}$이다. 따라서 f는 다음과 같은 이차식으로 나타낼 수 있다.

$$f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} = a_{11}x_1^2 + a_{22}x_2^2 + \cdots + a_{nn}x_n^2 + \sum_{i<j} 2a_{ij}x_i x_j$$

이제 f의 편도함수를 계산하면 다음과 같으므로

$$\frac{\partial f}{\partial x_k} = 2a_{k1}x_1 + 2a_{k2}x_2 + \cdots + 2a_{kn}x_n,\ k = 1, 2, \cdots, n$$

편도함수를 모두 구하여 하나의 벡터로 나타내면 f의 그래디언트는 다음과 같다.

$$\nabla f(\mathbf{x}) = \left[\sum_{i=1}^{n} 2a_{1i}x_i,\ \sum_{i=1}^{n} 2a_{2i}x_i,\ \cdots,\ \sum_{i=1}^{n} 2a_{ni}x_i\right]^T = 2A\mathbf{x}$$

그리고 $g(\mathbf{x}) = \mathbf{x}^T\mathbf{x} - 1 = \mathbf{x}^T(I\mathbf{x}) - 1$는 이차형식에 A대신 단위행렬 I가 있으므로 $\nabla g(\mathbf{x}) = 2I\mathbf{x} = 2\mathbf{x}$가 된다. 따라서 다음과 같은 관계를 얻는다.

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x}) \quad \Rightarrow \quad 2A\mathbf{x} = 2\lambda\mathbf{x} \quad \Rightarrow \quad A\mathbf{x} = \lambda\mathbf{x}$$

이렇게 $\mathbf{0}$이 아닌 벡터 $\mathbf{x} \in \mathbb{R}^n$에 대하여 $A\mathbf{x} = \lambda\mathbf{x}$을 만족하는 λ를 A의 고윳값이라 하고, 이때의 벡터 $\mathbf{x}$를 고윳값 λ에 대응하는 A의 고유벡터라고 한다. 즉 대응하는 행렬의 고윳값을 계산하는 과정은 바로 라그랑주의 승수를 구하는 것과 일치한다. 또한 고윳값 λ와 대응하는 고유벡터 $\mathbf{x}$에 대하여 $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x} = \mathbf{x}^T(\lambda\mathbf{x}) = \lambda(\mathbf{x}^T\mathbf{x}) = \lambda$이 성립하므로, 함수 f의 최댓값과 최솟값은 행렬 A의 가장 큰 실수 고윳값과 가장 작은 실수 고윳값이 됨을 쉽게 알 수 있다.

물론 선형대수학을 이수하지 않은 학생이라면 여기서 처음으로 행렬의 고윳값과 고유벡터를 접하게 될 것이다. 그리고 선형대수학을 수강한 학생이라고 하더라도 이차형식을 통해 미분적분학과 고윳값, 고유벡터 사이의 관계를 새롭게 조망해 볼 수 있는 기회를 가질 수 있을 것이다. 그리고 이와 같은 설명을 통해 고윳값 문제의 근본 개념을 깊이 고찰할 수 있을 것이다. 정사각행렬의 고윳값과 고유벡터는 아래의 코드로 계산할 수 있다. 이 간단한 코드는 n차 행렬의 고윳값과 고유벡터들을 찾는데 그대로 활용되므로 중요한 의미가 있다.

```
A = matrix(QQ, 3, 3, [1, 2, 2, 1, 2, -1, 3, -3, 0] )  # 3x3 행렬
# A = matrix(QQ, [[10, -3, 0, -3], [-6, -1, 6, -1], [4, -4, 6, -4], [6, 5, -6, 5]]) # 4x4 행렬 (행을 기준으로)
# A = matrix(QQ, n, n, [*, *, ... , *, *])  # nxn 행렬
print ( A.eigenvectors_right() )  # 고윳값, 고유벡터 계산
```

3. 라그랑주 승수법과 주성분 분석

최근 각광 받고 있는 인공지능을 활용하기 위해서는 관련 데이터의 분석이 필수적이다. 앞서 언급했듯이, 라그랑주 승수법은 미분적분학과 선형대수학을 연결 짓는 중요한 개념으로 이해할 수 있는데, 라그랑주 승수법의 응용 사례 중 주목할 만한 것이 빅데이터와 관련하여 널리 활용되는 주성분 분석(PCA)이다. PCA는 가장 널리 사용되는 차원 축소 기법 중 하나로, 원 데이터의 분포를 최대한 보존하면서도 고차원 데이터를 저차원 데이터로 변환한다. PCA는 통계적 관점에서 확률변수의 일차결합으로 만든 새로운 변수(주성분, principal component, PC)의 분산을 최대가 되도록 하는데, 이때 라그랑주 승수법이 적용된다(이상구·이재화, 2019).

PCA의 목표는 주어진 데이터로부터 계산한 공분산 행렬 Σ로부터 얻은 정보를 최대한 보존하면서도 훨씬 적은 개수의 새로운 변수($\ll p$)로 기존 데이터의 정보 대부분을 표현하려고 하는 것이다. 즉, 상관관계가 큰 변수들로 이루어진 데이터 집합의 차원을 축소하는 것이다. 이는 기하적으로 p차원 공간에서 데이터들의 분산을 최대화하는 새로운 직선(또는 평면)을 찾는 최적화 문제임이 알려져 있다(Pearson, 1901).

첫 번째 주성분을 찾기 위해서 센터링(centering, 평균을 0으로 조정) 된 확률변수 x_1, ..., x_p의 일차결합으로 만든 새로운 변수 $z_1 = \mathbf{v}_1^T\mathbf{x} = v_{11}x_1 + v_{21}x_2 + \cdots + v_{p1}x_p$를 생각해 보자. 여기서 $\mathbf{v}_1$은 p개의 상수 $v_{11}, v_{21}, \ldots, v_{p1}$을 성분으로 하는 벡터로 '적재계수(loadings)'라 부르기도 한다. 그러면 z_1의 분산은 $\mathrm{Var}(z_1) = \mathrm{Var}(\mathbf{v}_1^T\mathbf{x}) = \mathbf{v}_1^T\Sigma\mathbf{v}_1$과 같이 이차형식으로 표현된다. 이것만으로 최대를 찾는 것은 불가능하므로 추가적으로 $\mathbf{v}_1$을 정규화(normalization)하는 조건, 즉 $\mathbf{v}_1^T\mathbf{v}_1 = 1$인 조건을 부여하여 다음의 최적화 문제를 구성한다.

<표 III-4> 첫 번째 주성분을 찾는 문제

$$\begin{aligned} &\text{maximize } \mathrm{Var}(z_1) = \mathrm{Var}(\mathbf{v}_1^T\mathbf{x}) = \mathbf{v}_1^T\Sigma\mathbf{v}_1 \\ &\text{subject to } \mathbf{v}_1^T\mathbf{v}_1 = 1 \end{aligned}$$

그러면 <표 III-4>의 최적화 문제는 앞서 살펴본 행렬의 고윳값과 고유벡터를 찾는 문제와 같다. 특히 분산을 최대화하므로($\mathrm{Var}(z_1) = \mathrm{Var}(\mathbf{v}_1^T\mathbf{x}) = \mathbf{v}_1^T\Sigma\mathbf{v}_1 = \lambda_1$) 이 문제의 라그랑주 승수는 Σ의 가장 큰 고윳값(λ_1)이고, $\mathbf{v}_1$은 그에 대응하는 단위 고유벡터가 된다.

두 번째 주성분은 분산이 최대가 되도록 하면서도 $z_1 = \mathbf{v}_1^T\mathbf{x}$과는 상관관계가 없도록 하는 $z_2 = \mathbf{v}_2^T\mathbf{x}$를 찾아야 한다. 이 때문에 두 번째 주성분을 찾는 문제는 제약조건이 하나 더 늘어난다. 이는 <표 III-2>와 같이 제약조건이 2개인 최적화 문제이다.

<표 III-5> 두 번째 주성분을 찾는 문제

$$\begin{aligned} &\text{maximize} && \mathrm{Var}(z_2) = \mathrm{Var}(\mathbf{v}_2^T\mathbf{x}) = \mathbf{v}_2^T\Sigma\mathbf{v}_2 \\ &\text{subject to} && \mathbf{v}_2^T\mathbf{v}_2 = 1 \\ & && \mathbf{v}_2^T\mathbf{v}_1 = 0 \end{aligned}$$

여기서 $\mathbf{v}_2^T\mathbf{v}_1 = 0$이라는 제약조건은 z_1과 z_2의 상관관계가 없도록 만드는 것으로, 확률변수 z_1과 z_2의 공분산이 0이 되면 이런 조건을 만족한다. 즉, $\mathrm{Cov}[z_1,\ z_2] = \mathrm{Cov}[\mathbf{v}_1^T\mathbf{x},\ \mathbf{v}_2^T\mathbf{x}] = \mathbf{v}_1^T\Sigma\mathbf{v}_2 = \lambda\mathbf{v}_1^T\mathbf{v}_2 = 0$이면 해당 제약조건을 도출할 수 있다. 왜냐하면 공분산행렬 Σ는 정의에 의해 항상 양의 정부호(positive definite)이므로 고윳값 λ가 양수이기 때문이다. 첫 번째 주성분을 찾는 과정과 마찬가지로 라그랑주의 승수법을 사용할 수 있는데 이때, 두 개의 라그랑주 승수 λ와 ϕ가 필요하다.

라그랑주 함수를 도입하면 $\mathbf{v}_2^T\Sigma\mathbf{v}_2 - \lambda(\mathbf{v}_2^T\mathbf{v}_2 - 1) - \phi\mathbf{v}_2{}^T\mathbf{v}_1$이고, 변수들은 $\mathbf{v}_2$ 벡터의 성분이므로 라그랑주 함수의 그래디언트가 0이 되도록 하는 식은 $\Sigma\mathbf{v}_2 - \lambda\mathbf{v}_2 - \frac{1}{2}\phi\mathbf{v}_1 = \mathbf{0}$이다. 그리고 이 식의 양변에 $\mathbf{v}_1$을 내적하면 $\mathbf{v}_1^T\Sigma\mathbf{v}_2 - \lambda\mathbf{v}_1^T\mathbf{v}_2 - \frac{1}{2}\phi\mathbf{v}_1^T\mathbf{v}_1 = 0$을 얻는다. 이때 z_1과 z_2의 공분산이 0이어야 하므로 $\mathbf{v}_1^T\Sigma\mathbf{v}_2 = 0$, $\mathbf{v}_1^T\mathbf{v}_2 = 0$에서 $\phi = 0$이 되어야 한다. 따라서 $(\Sigma - \lambda I_p)\mathbf{v}_2 = \mathbf{0}$을 얻는다. 역시 분산을 최대화하므로 λ는 Σ의 두 번째로 큰 고윳

값(λ_2)이고, v_2는 그에 대응하는 단위 고유벡터가 된다.

같은 방법으로 k번째 주성분 $z_k = \mathrm{v}_k^T \mathbf{x}$의 분산이 최대가 되도록 하되 $z_1 = \mathrm{v}_1^T \mathbf{x}$, $z_2 = \mathrm{v}_2^T \mathbf{x}$, ..., $z_{k-1} = \mathrm{v}_{k-1}^T \mathbf{x}$와는 상관관계가 없도록 z_k를 찾는 최적화 문제는 다음과 같이 주어진다.

<표 III-6> k번째 주성분을 찾는 문제

maximize	$\mathrm{Var}(z_k) = \mathrm{Var}(\mathrm{v}_k^T \mathbf{x}) = \mathrm{v}_k^T \Sigma \mathrm{v}_k$
subject to	$\mathrm{v}_k^T \mathrm{v}_k = 1$
	$\mathrm{v}_k^T \mathrm{v}_1 = 0,\ \mathrm{v}_k^T \mathrm{v}_2 = 0,\ \ldots,\ \mathrm{v}_k^T \mathrm{v}_{k-1} = 0$

λ_k를 Σ의 k번째로 큰 고윳값, v_k를 그에 대응하는 단위 고유벡터라 하면, k번째 주성분 $z_k = \mathrm{v}_k^T \mathbf{x}$의 분산은 λ_k이다. 즉 $\mathrm{Var}(z_k) = \mathrm{Var}(\mathrm{v}_k^T \mathbf{x}) = \lambda_k$가 성립한다. 이런 식으로 다음 주성분을 찾기 위해 위의 과정을 반복해 나가면 제약조건이 점점 늘어나는 최적화 문제들이 등장한다. 물론 여기서 실제 라그랑주 승수법으로 공분산 행렬의 고윳값과 고유벡터를 계산할 필요는 없다. 앞서 소개한 이차형식의 최대, 최소 문제, 행렬의 고윳값, 고유벡터 및 라그랑주 승수법의 선형대수학적 해석을 통해 공분산 행렬의 고윳값 분해 또는 데이터 행렬의 특잇값 분해(SVD)로 계산할 수 있다. 그러나 원 데이터의 분포를 최대한 보존하면서도 데이터의 차원을 줄이고자 하는 PCA의 원리를 이해하기 위해서는 $\mathrm{v}_1^T \mathrm{v}_1 = 1$에 대하여 주성분 $z_1 = \mathrm{v}_1^T \mathbf{x}$의 분산 $\mathrm{Var}(\mathrm{v}_1^T \mathbf{x}) = \mathrm{v}_1^T \Sigma \mathrm{v}_1$이 최대가 되도록 다음과 같이 라그랑주 함수

$$L(\mathrm{v}_1, \lambda) = \mathrm{v}_1^T \Sigma \mathrm{v}_1 - \lambda(\mathrm{v}_1^T \mathrm{v}_1 - 1)$$

의 v_1에 관한 그래디언트를 $\mathbf{0}$으로 두어 관계식 $2(\Sigma \mathrm{v}_1 - \lambda \mathrm{v}_1) = \mathbf{0}$, 즉 $\Sigma \mathrm{v}_1 = \lambda \mathrm{v}_1$을 만족해야 함을 이해해야 한다.

라그랑주 승수법의 응용 사례로, 앞서 제시한 교수학습자료로 수강했던 학생들이 PCA를 활용하여 실제 데이터를 분석한 결과를 소개한다. 이 데이터[2)]에는 유럽축구연맹 선수들의 크로스(cross) 횟수부터 골키퍼의 민첩성(GK Reflexes)까지 33개의 특성이 정리되어 있다([그림 Ⅲ-3]). 예시가 잘 드러날 수 있도록 만여 명의 선수들에 대한 빅데이터 중 개인용 PC에서 무리 없이 분석 결과를 얻어낼 수 있도록 첫 100개의 데이터만 사용하기로 하고, 파이썬 언어 대신 통계 분야에서 주로 사용하는 R을 활용하여 PCA를 진행하였다. 사용한 R 코드는 <표 Ⅲ-7>과 같다. 이 코드를 복사하여 웹 사이트 https://sagecell.sagemath.org/에 붙여넣고 오른쪽 하단의 언어 선택에서 R 언어를 선택하여 시행하면 결과(output)를 확인할 수 있다. 다른 데이터가 주어져도 이 코드를 활용하여 분석할 수 있다.

2) https://raw.githubusercontent.com/twackycats/european-soccer-analysis/master/data/player.attributes.csv

	crossing	finishing	heading_accuracy	short_passing	volleys	dribbling	curve	free_kick_accuracy	long_passing	ball_control	acceleration	sprint_speed	agility
1	49	44	71	61	44	51	45	39	64	49	60	64	59
2	49	44	71	61	44	51	45	39	64	49	60	64	59
3	49	44	71	61	44	51	45	39	64	49	60	64	59
4	48	43	70	60	43	50	44	38	63	48	60	64	59
5	48	43	70	60	43	50	44	38	63	48	60	64	59
6	80	53	58	71	40	73	70	69	68	71	79	78	78
7	80	53	58	71	32	73	70	69	68	71	79	78	78
8	79	52	57	70	29	71	68	69	68	70	79	78	78
9	79	51	57	70	29	71	68	69	68	70	79	78	78
10	79	51	57	70	29	71	68	69	68	70	79	78	78
11	79	51	57	70	29	71	68	69	68	70	79	78	78
12	79	51	57	70	29	71	68	69	68	70	80	78	78
13	79	51	57	70	29	67	68	69	68	68	79	78	78
14	78	50	56	69	28	66	67	68	67	67	79	82	79
15	78	50	56	69	28	66	67	68	67	67	79	82	79
16	78	50	56	69	28	66	67	68	67	67	79	82	79

[그림 Ⅲ-3] 유럽축구연맹 선수들의 특성 데이터 예시

<표 Ⅲ-7> PCA를 수행하는 R 코드

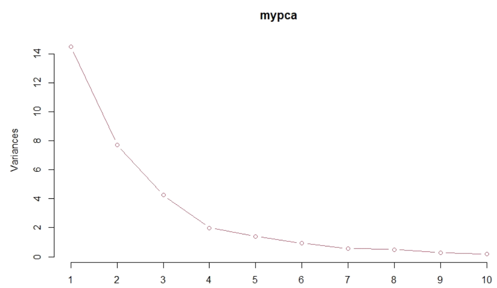
```
# R 프로그램에서 아래 명령어를 실행하면 다음 결과들을 얻을 수 있다.
file <- read.csv
('https://raw.githubusercontent.com/twackycats/european-soccer-analysis/master/data/player.attributes.csv',
fileEncoding="utf-8")
data <- file[c(1:100),-c(1:10)]
mypca <- prcomp(data, center=TRUE, scale=TRUE, retx=TRUE)
summary(mypca)
# scree 그래프 (분산을 나타낸다)
screeplot(mypca, type="lines", col = 2)
# 2차원으로 차원 축소 후 PC1과 PC2 사이의 관계를 살펴본다.
biplot(mypca)
# 3차원으로 차원 축소 후 PC1과 PC2, PC3만 취하여 k-means clustering 진행한다.
# PC1, PC2, PC3를 이용하여 클러스터의 개수(centers)를 4개로 하고
# 임의로 100번 시행(nstart)하여 가장 잘 clustering 하는 것을 찾는다.
myclust <- kmeans(mypca$x[,1:3], centers=4, nstart=100)
# clustering 후 시각화 결과를 보여준다.
# install.packages("rgl")
library(rgl)
plot3d(mypca$x[,1:3], xlab="PC 1", ylab="PC 2", zlab="PC 3", cex=1.5, size=1, type="s", col=myclust$cluster)
text3d(myclust$centers, texts=c("CL 1","CL 2","CL 3","CL 4"))
```

[그림 Ⅲ-4]는 PCA 실행 결과를 요약한 것으로, 만일 라그랑주 승수가 공분산 행렬의 고윳값, 즉 주성분의 분산이 된다는 것을 이해하지 못하면, 명령어로 얻은 결과를 그냥 받아들일 수 밖에는 없다. 그러나 라그랑주 승수와 주성분 분석 사이의 관계를 알고 있으면, 쉽게 코드의 내용과 결과를 이해할 수 있다. 예를 들어, [그림 Ⅲ-4]에서 표준편차(standard deviation) 행의 값들은 각 주성분의 표준편차를 의미하므로 이 값들을 제곱한 것이 바로 공분산 행렬의 고윳값이 된다. 그리고, 고윳값들의 전체 합에 대한 각각의 고윳값들의 비율이 해당 주성분이 데이터의 분포를 보존하는 비율이 된다. 따라서 분산의 누적 비율(Cumulartive Proportion)로부터 PC1, PC2가 원 데이터의 분포를 약 67% 보존하고, PC3까지 고려하면 약 80% 정도를 보존한다고 이해할 수 있다.

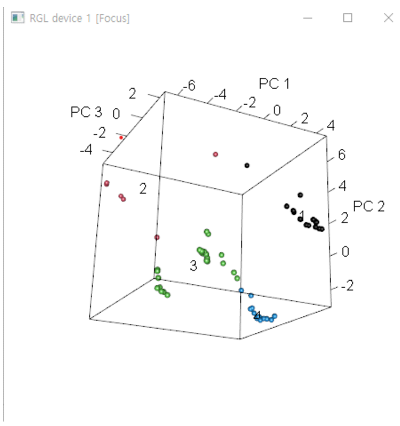
```
> mypca <- prcomp(data, center=TRUE, scale=TRUE, retx=TRUE)
> summary(mypca)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11    PC12
Standard deviation     3.8052 2.7762 2.0589 1.40997 1.18514 0.95907 0.75378 0.69995 0.53314 0.44175 0.41384 0.37410
Proportion of Variance 0.4388 0.2336 0.1285 0.06024 0.04256 0.02787 0.01722 0.01485 0.00861 0.00591 0.00519 0.00424
Cumulative Proportion  0.4388 0.6723 0.8008 0.86103 0.90360 0.93147 0.94869 0.96353 0.97215 0.97806 0.98325 0.98749
                          PC13    PC14    PC15   PC16    PC17    PC18    PC19    PC20    PC21    PC22    PC23   PC24
Standard deviation     0.31203 0.26771 0.21643 0.1908 0.17431 0.16710 0.15747 0.14361 0.12420 0.10428 0.09078 0.0805
Proportion of Variance 0.00295 0.00217 0.00142 0.0011 0.00092 0.00085 0.00075 0.00062 0.00047 0.00033 0.00025 0.0002
Cumulative Proportion  0.99044 0.99261 0.99403 0.9951 0.99606 0.99690 0.99765 0.99828 0.99875 0.99908 0.99933 0.9995
                          PC25    PC26    PC27    PC28    PC29    PC30    PC31    PC32    PC33
Standard deviation     0.07207 0.06075 0.05668 0.03524 0.03026 0.02593 0.01996 0.01587 0.01397
Proportion of Variance 0.00016 0.00011 0.00010 0.00004 0.00003 0.00002 0.00001 0.00001 0.00001
Cumulative Proportion  0.99968 0.99979 0.99989 0.99993 0.99995 0.99997 0.99999 0.99999 1.00000
```

[그림 III-4] PCA 실행 결과 요약

데이터를 몇 차원으로 줄일지 결정하기 위해 주성분의 분산(즉 공분산 행렬의 고윳값) 변화를 선 그림(scree plot)으로 그려보았다([그림 Ⅲ-5]). 대부분 기울기가 꺾이는 ‘elbow point’의 왼쪽에 있는 성분까지 선택하는 데 여기서는 3번째 주성분까지 선택하였다. 따라서 20% 정도의 정보는 잃어버리더라도 3차원으로 데이터의 차원을 축소할 수 있다. 그러면 데이터를 시각화하여 분석하기가 용이하다. 또한 차원 축소된 데이터에 k-means clustering과 같은 다른 데이터 분석 기법을 적용할 수도 있다. 실제로 [그림 Ⅲ-6]은 3차원으로 차원 축소된 데이터를 4개의 군집으로 시각화하였다.

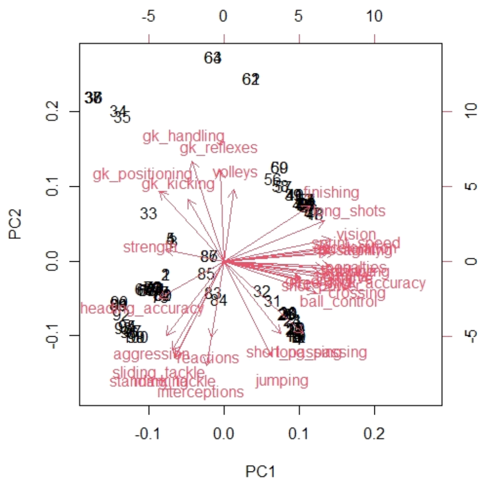


[그림 III-5] 주성분의 분산 변화를 나타낸 선 그림



[그림 III-6] 3차원으로 차원 축소된 데이터를 4개의 군집으로 시각화한 그림

[그림 Ⅲ-7]은 원 데이터의 변수와 주성분 사이의 관계를 나타낸 행렬도(biplot)로 주성분 PC1, PC2와 2차원으로 축소된 데이터, 그리고 원 데이터의 변수들을 모두 하나의 그림으로 나타낸 것이다. 일반적으로 각 주성분들과 평행에 가까운 속성일수록 적재계수의 절댓값이 크고, 양의 방향으로 향하는 화살표들은 양의 적재계수를 가지며, 음으로 향하는 화살표들은 음의 적재계수를 갖는다. 그리고 화살표들이 가까울수록 각 속성 간의 상관관계가 높다. 이를 통해 각 주성분에 큰 영향을 미치는 속성들이 무엇인지 파악할 수 있다.



[그림 Ⅲ-7] 원 데이터의 변수와 주성분 사이의 관계를 나타낸 행렬도(biplot)

4. 등식 제약조건이 있는 최적화 문제의 1계 최적 조건에 대한 증명

이제 등식 제약조건이 있는 최적화 문제(<표 Ⅲ-8>)의 1계 최적 조건에 대한 증명을 소개한다. 이는 심화된 내용을 학습하기를 원하는 학생들에게 적절한 난이도로 제공할 수 있다.

<표 Ⅲ-8> 여러 개의 등식 제약조건이 있는 최적화 문제(독립변수가 n개인 경우)

minimize (maximize)	$f(\mathbf{x})$	$\mathbf{x} \in \mathbb{R}^n$
subject to	$c_i(\mathbf{x}) = 0,$	$i = 1, \ldots, m$

이제 $\mathbf{x}^*$를 위의 문제의 최적해라 하면 다음 조건 ①을 만족하는 모든 $\mathbf{d}$에 대하여 조건 ②가 성립해야 한다.

$$\mathbf{d}^T \nabla c_i(\mathbf{x}^*) = 0, \quad i = 1, 2, \ldots, m \quad \text{— ①}$$
$$\mathbf{d}^T \nabla f(\mathbf{x}^*) = 0 \quad \text{— ②}$$

즉 $\mathbf{x}^*$에서 $\mathbf{d}$ 방향으로 이동해도

① 제약조건 $c_i(\mathbf{x}) = 0 \ (i = 1, \ldots, m)$를 모두 만족하며,

② 함수 f의 순간변화율이 0이라야 한다.

따라서 $\nabla c_i(\mathbf{x}^*)$, $i=1,\dots,m$가 일차독립이면, $\nabla f(\mathbf{x}^*)=\sum_{i=1}^{m}\lambda_i \nabla c_i(\mathbf{x}^*)$을 만족하는 적당한 상수 λ_i^*, $i=1,\dots,m$가 존재해야 한다. 이를 바탕으로 위의 최적화 문제를 푸는 라그랑주 승수법은 다음과 같이 제시된다.

[단계 1] 다음 연립방정식을 만족하는 $\mathbf{x}$, λ_i $(i=1,\dots,m)$의 값을 모두 구한다.

$$\nabla f(\mathbf{x})=\sum_{i=1}^{m}\lambda_i \nabla c_i(\mathbf{x}),\quad c_i(\mathbf{x})=0,\ i=1,\dots,m$$

[단계 2] [단계 1]에서 구한 모든 점 $\mathbf{x}$에서 f의 값을 계산한다. 이 중에서 가장 큰 값이 f의 최댓값이고, 가장 작은 값이 f의 최솟값이다.

이때, 다음과 같이 라그랑주 함수를 정의하면

$$L(\mathbf{x},\lambda_1,\dots,\lambda_m)=f(\mathbf{x})-\lambda_1 c_1(\mathbf{x})-\cdots-\lambda_m c_m(\mathbf{x})=f(\mathbf{x})-\sum_{i=1}^{m}\lambda_i c_i(\mathbf{x})$$

라그랑주 승수법의 [단계 1]은 결국 $\nabla L(\mathbf{x},\lambda_1,\dots,\lambda_m)=\mathbf{0}$가 되는 $\mathbf{x}$, λ_i $(i=1,\dots,m)$를 찾는 것과 같다.

5. 라그랑주 승수와 카루시-쿤-터커(Karush - Kuhn - Tucker, KKT) 조건

라그랑주 승수법은 경제학에서도 많이 사용된다. 예를 들어, 정해진 한도가 있는 예산의 제약 아래 기업의 이익을 최대화, 생산 비용을 최소화, 또는 소비자의 효용을 최대화하는 등의 행동을 분석하는 수학적 모형이 최적화 문제로 나타나는데, 이때 라그랑주 승수법이 활용된다. 게다가 경제학에서 외생변수(exogenous variable)[3]의 변화에 따른 평형상태(equilibrium)의 변화를 분석하는 비교정태분석(comparative statics)을 수행할 때, 라그랑주 승수 자체가 큰 의미를 가진다. 예를 들어, 예산의 제약조건이 변화하는 경우 경제 주체들의 경제적 행동도 변화하는데, 이는 수학적 모형의 라그랑주 승수의 의미를 파악함으로써 분석할 수 있다.

라그랑주 승수의 의미를 간단히 설명하기 위해 목적함수가 $f(x,y)$인 2변수 함수이고, 제약조건이 $g(x,y)=c$ 또는 $g(x,y)-c=0$으로 주어지는 경우를 생각해보자. 즉 우변이 0에서 c로 변화하였다. 여기서 c를 새로운 매개변수(parameter)라고 생각할 수 있다. 이때 라그랑주 함수는 $L(x,y,\lambda)=f(x,y)-\lambda(g(x,y)-c)$이 되고, L의 그래디언트는 $\nabla L=(f_x-\lambda g_x,\ f_y-\lambda g_y,\ c-g)$가 된다.

만일 $\nabla L=\mathbf{F}$라 하면, 라그랑주 승수법은 $\nabla L(x,y,\lambda)=\mathbf{0}$, 즉 방정식 $\mathbf{F}(x,y,\lambda;c)=\mathbf{0}$을 푸는 것이 된다. 따라서 방정식이 음함수 $x=x(c)$, $y=y(c)$, $\lambda=\lambda(c)$를 정의한다고 할 수 있다. 만약 $\mathbf{F}$에 대해 변수 x, y, λ에 대한 야코비안 행렬식 $|J_{\mathbf{F}}|$이 0만 아니라면 음함수 정리에 의해 이는 사실일 뿐만 아니라 $x(c)$, $y(c)$, $\lambda(c)$는 미분가능하고 그 도함수들은 연속이다. 그리고 실제 응용에서는 경험적으로 $|J_{\mathbf{F}}|\neq 0$이라는 가정이 합리적이다.

이제 목적함수 $f(x(c),y(c))$를 c로 미분하면 연쇄법칙에 의해 다음과 같은 식을 얻는다.

$$\frac{df(x(c),y(c))}{dc}=\frac{\partial f}{\partial x}\frac{dx}{dc}+\frac{\partial f}{\partial y}\frac{dy}{dc}$$

---

[3] 독립변수 이외의 변수로서 종속변수에 영향을 주어 이를 통제하지 않으면 연구 결과의 내적 타당도가 문제가 되는 변수

여기서 $\frac{\partial f}{\partial x}=\lambda\frac{\partial g}{\partial x}$, $\frac{\partial f}{\partial y}=\lambda\frac{\partial g}{\partial y}$ 이므로 이를 위의 식에 대입하면

$$\frac{df(x(c),y(c))}{dc}=\lambda\frac{\partial g}{\partial x}\frac{dx}{dc}+\lambda\frac{\partial g}{\partial y}\frac{dy}{dc}=\lambda\left(\frac{\partial g}{\partial x}\frac{dx}{dc}+\frac{\partial g}{\partial y}\frac{dy}{dc}\right)$$

이다. 그리고 제약조건의 양변을 c로 미분하면

$$\frac{dg(x(c),y(c))}{dc}=\frac{\partial g}{\partial x}\frac{dx}{dc}+\frac{\partial g}{\partial y}\frac{dy}{dc}=1=\frac{d}{dc}(c)$$

이 되고, 이를 위의 식에 대입하면 다음을 얻는다.

$$\frac{df}{dc}=\lambda$$

따라서 라그랑주 승수는 제약조건 $g(x,y)=c$ 에서 매개변수 c의 변화가 목적함수의 최댓값 또는 최솟값에 미치는 영향의 정도를 나타낸다고 볼 수 있다.

라그랑주 승수법 중 등식 제약조건이 있는 최적화 문제의 1계 최적 조건은 아래와 같이 부등식 제약조건을 포함하는 일반화된 최적화 문제(<표 Ⅲ-9>)로 확장할 수 있다.

<표 Ⅲ-9> 등식 및 부등식 제약조건이 있는 최적화 문제

minimize (maximize)	$f(\mathbf{x})$	$\mathbf{x}\in\mathbb{R}^n$
subject to	$c_i(\mathbf{x})=0,$	$i=1,\ldots,m_e$
	$c_i(\mathbf{x})\ge 0,$	$i=m_e+1,\ldots,m$

여기서 $f,\ c_i:\mathbb{R}^n\to\mathbb{R}\ (i=1,\ldots,m)$는 모두 미분가능한 함수이며, $c_i(\mathbf{x})\ge 0\ (i=m_e+1,\ldots,m)$는 부등식 제약조건(inequality constraint)이라 한다. 앞서 살펴본 1계 최적 조건과 유사한 방법으로 <표 Ⅲ-9> 문제에 대한 최적 조건인 '카루시-쿤-터커(Karush - Kuhn - Tucker, KKT) 조건'을 증명할 수 있다. 이렇게 일반적인 최적화 문제는 현재까지도 연구가 활발한 수학 문제 중 하나이다.

정리 2. $\mathbf{x}^*$가 <표 Ⅲ-9> 문제의 최적해라 하자. 만일 $\mathbf{x}^*$에서 등식 $c_i(\mathbf{x}^*)=0$으로 만족되는 모든 제약조건(등식, 부등식 포함)에 대하여 $\nabla c_i(\mathbf{x}^*)$가 일차독립이면, 다음을 만족하는 적당한 상수 λ_i^*, $i=1,\ldots,m$가 존재한다.

$$\nabla f(\mathbf{x}^*)=\sum_{i=1}^{m}\lambda_i^*\nabla c_i(\mathbf{x}^*)$$
$$c_i(\mathbf{x}^*)=0,\quad i=1,\ldots,m_e$$
$$c_i(\mathbf{x}^*)\ge 0,\quad \lambda_i^*\ge 0,\quad \lambda_i^* c_i(\mathbf{x}^*)=0,\ i=m_e+1,\ldots,m$$

고등학교 미분적분학에서 배운 함수의 최대, 최소를 구하는 방법에서 발전한 라그랑주 승수법이 미분적분학

과 선형대수학을 연결하는 중요한 도구이며, 주성분 분석을 포함한 인공지능 알고리즘에 적극적으로 활용되고 있으며, 관련 연구가 지금도 활발하게 진행되고 있다는 것을 본 원고에서 확인하였다. 교수자는 처음 라그랑주 승수법을 접하는 학생들에게 앞서 제시한 교수학습자료를 활용하여 충분한 학습 동기를 제공할 수 있기를 기대한다. 특히 제공된 시각화 자료와 SageMath 코드는 앞으로 만날 다양한 실제 문제를 해결하는데 바로 적용이 가능할 것이다. 본 연구는 대학수학을 지도하는 다양한 전공의 교수자들에게 도움이 될 기초자료가 될 것이다.

Ⅳ. 결론 및 제언

본 논문에서는 자연과학, 공학, 사회과학 뿐만 아니라 기계학습, 인공지능 등에서 발생하는 최적화 문제들에 필수적인 라그랑주 승수법에 대하여 살펴보았다. 특히 대학 미분적분학에서 처음 다뤄지는 라그랑주 승수법의 교수학습에서 나타나는 한계를 극복하기 위해, 학생들에게 라그랑주 승수법을 교육할 때 교수자에게 도움이 될 필요한 지식을 종합하여 제시하였다.

먼저 라그랑주 승수법을 직관적으로 이해할 수 있도록 시각화 자료를 제시하였고, 계산에 대한 부담을 덜고 라그랑주 승수법의 원리에 집중할 수 있도록 파이썬 기반의 SageMath 코드를 제공하였다. 또한 라그랑주 승수법이 행렬의 고윳값과 고유벡터를 유도하는 과정을 상세히 설명하여, PCA와 같은 인공지능 알고리즘의 본질을 꿰뚫어 볼 수 있도록 하였다. 특히 실제 사례로 PCA를 학습한 학생들이 라그랑주 승수법과 고윳값을 이용하여 실제 데이터를 분석한 결과도 소개하였다. 그리고 더 심화학습을 원하는 학생들에게 참고자료로 제시할 수 있도록 라그랑주 승수법의 바탕이 되는 최적화 문제의 1계 최적 조건의 증명과 이를 확장한 내용도 소개하였다.

라그랑주 승수법은 역사적으로 오래된 방법이지만 현재에도 다양한 분야에서 지속적으로 활용되고 있다. 따라서 대학 미분적분학 강의에서 라그랑주 승수법의 계산적인 측면뿐만 아니라 다른 과목과의 연결성 및 활용사례, 코드 등이 제공된다면, 학생들이 라그랑주 승수법을 왜 배우는지, 라그랑주 승수가 의미하는 것이 무엇인지 깊게 이해하여 현실문제에 바로 적용할 수 있을 것이다. 본 원고에서 소개된 내용은 심화 교육에 사용될 수 있을 뿐만 아니라, 관련 교재 집필 등에 적절하게 활용할 수 있을 것이다.

감사의 글: 본 원고의 초안을 검토하고 윤문해 주신 함윤미 교수님, 김응기 박사님, 박경은 박사님께 감사드립니다.

참 고 문 헌

서보억 (2021). 구체적 수학탐구활동 사례를 통한 학교현장 수학 탐구방법 탐색, 수학교육논문집, **35(2)**, 193-212.

Suh, B.E. (2021). A study on mathematical investigation activity through using one mathematical fact. *Communications of Mathematical Education*, **35(2)**, 193-212.

이상구・이재화 (2019). [빅북] 인공지능을 위한 기초수학. 교보문고.

Lee, S.-G., & Lee, J.H.. (2019). *[BigBook] Basic Mathematics for Artificial Intelligence.* Kyobo Book Centre. http://matrix.skku.ac.kr/math4ai/Math4AI.pdf

이상구・이재화・유주연・함윤미 (2022). 다변수 미적분학 & 코딩. 경문사.

Lee, S.-G., Lee, J.H., Yoo, J.Y. & Ham, Y. (2022). *Multivariable Calculus & Coding.* Kyung Moon Sa. https://buk.io/@kc7895

Bussotti, P. (2003). On the genesis of the Lagrange multipliers, *Journal of Optimization Theory and*

Applications, 117(3), 453-459. https://doi.org/10.1023/A:1023952102705

Chiang, A.C. & Wainwright, K. (2004). *Fundamental Methods of Mathematical Economics (4th ed)*, McGraw-Hill.

Horn, R.A. & Johnson, C.R. (2013). *Matrix Analysis (2nd ed.),* Cambridge University Press.

Lee, S.-G., Kim, E.-K., Ham, Y., Kumar, A., Beezer, R., Vu, Q.-P., Hwang, S.-G. & Simon, L. (2014). *Calculus*. Kyung Moon Sa.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* 2(11), 559-572.

A Study on Teaching the Method of Lagrange Multipliers in the Era of Digital Transformation

Lee, Sang-Gu
Department of Mathematics, Sungkyunkwan University
E-mail : sglee@skku.edu

Nam, Yun
Institute of Basic Science, Sungkyunkwan University
E-mail : yunnam@skku.edu

Lee, Jae Hwa†
Convergence Research Center for Energy and Environmental Sciences, Sungkyunkwan University
E-mail : jhlee2chn@skku.edu

The method of Lagrange multipliers, one of the most fundamental algorithms for solving equality constrained optimization problems, has been widely used in basic mathematics for artificial intelligence (AI), linear algebra, optimization theory, and control theory. This method is an important tool that connects calculus and linear algebra. It is actively used in artificial intelligence algorithms including principal component analysis (PCA). Therefore, it is desired that instructors motivate students who first encounter this method in college calculus.

In this paper, we provide an integrated perspective for instructors to teach the method of Lagrange multipliers effectively. First, we provide visualization materials and Python-based code, helping to understand the principle of this method. Second, we give a full explanation on the relation between Lagrange multiplier and eigenvalues of a matrix. Third, we give the proof of the first-order optimality condition, which is a fundamental of the method of Lagrange multipliers, and briefly introduce the generalized version of it in optimization. Finally, we give an example of PCA analysis on a real data. These materials can be utilized in class for teaching of the method of Lagrange multipliers.

---

* 2000 Mathematics Subject Classification : 97U50, 97U60, 97U70
* Key words : method of Lagrange multipliers, calculus, linear algebra, optimization, control thoery, principal component analysis, quadratic form, Karush-Kuhn-Tucker (KKT) conditions
* This research was supported by Korea Initiative for fostering University of Research and Innovation Program of the National Research Foundation (NRF) funded by the Korean government (MSIT) (No.2020M3H1A1077095) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2021R1F1A1046714).
† Corresponding author