Selecting the Number and Location of Knots for Presenting Densities

안정용[1], 문길성[2], 한경수[3]

요 약

본 연구에서는 연속형 확률밀도함수의 그래프를 표현하기 위한 하나의 방법으로 보간점을 이용하는 문제에 대해 살펴보고자 한다. 이를 위해 최적화 기법을 이용하여 보간점의 수와 위치를 선택하는 알고리즘을 제안하고, 제안한 방법을 이용하여 확률밀도함수의 그래프를 구현한다.

주요용어 : 보간점, 스플라인, 베지어 곡선

1. 서론

통계 그래프는 데이터나 복잡하게 정의된 함수들의 형태에 대한 직관적인 이해를 돕기 위해 자주 사용되며, Wilkinson(1999)은 전문가와 비전문가 사이의 원활한 의사소통을 지원하는 연결고리로서 그 중요성을 강조하고 있다. 예를 들어, 연속형 확률변수의 분포를 표현하기 위해 사용되는 확률밀도함수의 대부분은 복잡한 수식으로 정의되며, 확률밀도함수를 보고 확률변수의 분포 모양을 상상하기란 쉬운 일이 아니다. 특히, 카이제곱분포나 F분포처럼 복잡한 수식으로 구성된 경우에 수작업을 통해 그래프를 표현하는 일은 쉽지 않으며, 가능하다해도 많은 노력과 시간이 소요된다. 따라서 이러한 함수들의 그래프를 표현하고자 할 때는 컴퓨터의 사용이 필수적이다.

상업용이나 공개용으로 개발된 통계 소프트웨어의 경우 확률밀도함수의 그래프를 그리기 위해 구간을 설정한 다음 주어진 구간을 등간격으로 나누고 각각의 좌표를 계산하여 선으로 연결하는 방법을 주로 사용한다. 그러나 이러한 방법은 함수 형태가 복잡할 경우 계산 시간이 많이 걸리게 되고 그래프의 형태도 부드럽게 연결되지 않는다는 단점을 가지고 있다.

컴퓨터에서 그래프를 표현하기 위해 점 또는 선으로 연결하는 방법은 구간내에서 함수가 선형으로 근사하는 것을 가정한다. 선형적인 근사 방법은 몇 가지 문제점을 가지고 있는데 그 중에서 대표적인 문제점은 표현하고자 하는 함수의 그래프는 연속인 반면 픽셀(pixel)로 이루어진 모니터는 이산적인 표현만 가능하다는 것이다. 따라서 좀 더 부드러운 형태의 그래프를 그리기 위해서는 점의 수를 늘리거나 연결하는 선의 간격을 작게 해야 한다. 그러나 몇 개의 점을 그릴 것인지, 몇 개의 구간으로 나누어 직선으로 연결할 것인지를 결정하는 것은 모니터가 지원하는 해상도의 영향을 받기 때문에 점의 수와 구간의 수는 상대적으로 변해야 한다. 또한 곡률 변화가 큰 부분에서는 점을 찍거나 선을 연결하는 방식은 제대로 곡선을 근사시킬 수 없다.

이러한 문제점을 보안하기 위한 대안으로 점이나 선이 아닌 곡선을 이용한 근사를 이용할 수 있다. Wegman과 Carr(1993)는 부드러운 곡선을 표현하기 위해서 3차 스플라인(cubic spline), B-스플라인(B-spline), 베지어(Bezier) 등 여러 가지 보간과 근사 방법을 제시하였다. 이러한 곡

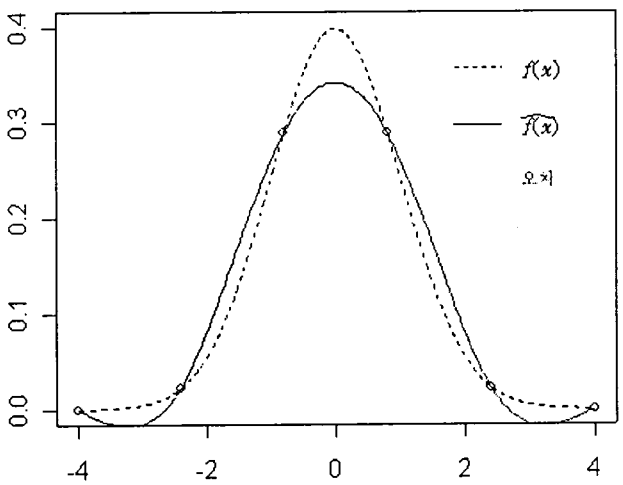
1) 전북대학교 수학·통계정보과학부 조교수, 전주시 덕진구 덕진동 664-14
2) 전북대학교 통계정보학과 박사과정, 전주시 덕진구 덕진동 664-14
3) 전북대학교 수학·통계정보과학부 교수, 전주시 덕진구 덕진동 664-14

선들을 이용할 때 좀더 부드럽게 곡선을 근사시킬 수 있다. 그러나 곡선으로 연결하는 방법은 Kim과 Lee(2001)가 지적한 바와 같이 보간점의 수와 배치에 따라 근사하는 정도 차이가 크다는 단점을 가지기 때문에 보간점의 배치가 중요하다. 따라서 효율적으로 곡선을 근사하기 위해 적절한 보간점의 수와 위치를 결정하는 방법이 필요하다.

본 연구에서는 확률밀도함수의 그래프를 곡선으로 근사할 때 적절한 보간점의 수와 위치를 선택하기 위한 방법으로 최적화 기법을 이용하는 알고리즘을 제안하고, 제안한 방법을 이용하여 확률밀도함수의 그래프를 구현하고자 한다.

2. 보간점의 수와 위치 선택

폐구간에서 정의된 임의의 함수를 근사하기 위한 하나의 방법으로 고차 다항식을 이용한 근사가 있다. 그러나 고차 다항식의 진동 성질과 구간의 한 부분에서의 파동이 전체 영역에 걸쳐서 큰 파동을 일으킨다는 사실은 다항식의 이용을 제한한다. 이에 대한 대안으로 주어진 구간을 부분구간들의 모임으로 나누고 각 부분구간에 대한 근사 다항식을 만드는 것을 고려할 수 있다. 이 같은 근사 방법을 부분구간 근사 다항식(piecewise polynomial approximation)이라 하며, 3차 스플라인 보간법이 대표적인 예이다. 그러나 이 방법은 같은 수의 보간점을 이용하더라도 보간점의 배치에 따라 근사 정도의 차이가 심하다는 단점을 가지고 있다. 따라서 곡선의 기하학적인 특성을 잘 반영하기 위해서는 곡률의 변화가 큰 부분에 작은 부분보다 많은 보간점이 필요하다.



<그림 1> 근사한 곡선의 오차

적당한 보간점의 배치를 찾는 문제는 <그림 1>과 같이 원래의 함수와 근사 함수간의 최대 오차를 최소로 하는 배치를 찾는 문제와 동일하다. 최적화 기법은 이러한 문제를 해결하기 위한 기법이며, 미분 정보를 이용하는 방법과 이용하지 않는 방법이 있다. 오차를 최소로 하는 함수에서는 미분을 할 수 없기 때문에 미분 정보를 필요로 하지 않는 최적화 기법을 이용해야 한다. 미분 정보가 필요하지 않는 최적화 기법으로는 다운힐 심플렉스(Downhill Simplex) 방법과 Simulated Annealing 방법이 있다. Simulated Annealing 방법은 확률적으로 전체적인 최소값을 찾아가는 방법으로 다운힐 심플렉스 방법에 비해 느리다(Press 등, 1992). 따라서 본 연구에서는 최적화 기법으로 다운힐 심플렉스 방법을 이용하여 보간점의 수와 위치를 선택하는 다음과

같은 알고리즘을 사용한다.

algorithm

1. 목적함수를 정의한다

$$g(f,\hat{f}) = \sup_{\Omega} |f(x) - \widehat{f(x)}|$$

2. 근사 구간을 결정한다
3. 보간점 수의 초기값을 결정한다
4. 보간점의 수, 최대오차, 보간점을 구한다
5. 오차의 임계값을 결정한다
6. 적절한 보간점의 수와 위치를 선택한다

<표 1> 표준정규분포에서의 보간점의 수, 최대오차, 보간점

보간점의 수	최대오차	보간점
4	0.1146209	-3.030032, -0.8918424, 0.8918424, 3.030032
5	0.00121699	-3.413662, -1.515855, 0, 1.515855, 3.413662
6	0.000972033	-3.431835, -1.492110, -0.1062158, 0.1062158, 1.492110, 3.431835]
7	0.001214872	-3.39638, -1.695623, -1.503918, 0, 1.503918, 1.695623, 3.39638
8	0.000958947	-3.434928, -2.427919, -1.496361, -0.08955839, 0.08955839, 1.496361, 2.427919, 3.434928

<표 2> 카이제곱분포의 보간점 수에 따른 최대오차(자유도:1, 30)

보간점의 수	최대오차
4	0.6034576
5	0.1268449
6	0.2017526
7	0.04049637
8	0.1035869
9	0.0665777
10	0.04072025
11	0.03913196
12	0.02972236
13	0.02039262
14	0.03571033
15	0.02422591
16	0.03911968
17	0.01459762
18	0.04032351
19	0.01598744
20	0.004871932

보간점의 수	최대오차
4	0.008169031
5	0.008170976
6	0.000258807
7	0.00017358
8	0.000197622
9	0.0000966

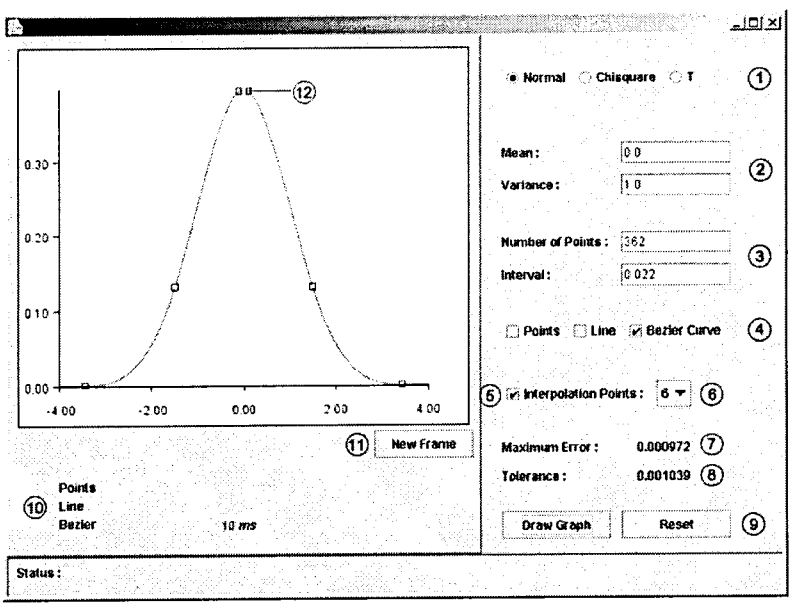
<표 1>은 표준정규분포의 확률밀도함수를 근사하기 위해 위의 알고리즘을 이용하여 구한 보간점의 수와 최대오차를 정리한 표이다(임계값: 0.001, 모니터 해상도: 1024*768 가정). 보간점의 수가 6개와 8개일 때 임계값보다 작은 최대오차를 갖는데 보간점의 수가 많아지면 많아질수록

근사 다항식이 복잡해지기 때문에 적은 수의 보간점을 선택한다. <표 2>는 자유도가 각각 1, 30인 카이제곱분포의 보간점의 수에 대한 최대오차를 나타낸 표(임계값: 0.01, 0.0001)이며, 자유도에 따라 근사에 필요한 보간점의 수가 다르다는 것을 보여주고 있다.

3. 그래프 구현

본 연구에서 확률밀도함수의 그래프는 앞절에서 제안한 알고리즘으로 구한 보간점을 이용한다. 이 보간점을 3차 스플라인 보간법으로 근사시키고, 이를 베지어 곡선으로 변환하여 구현하였다. 베지어 곡선은 프로그램 언어에서 곡선을 그리기 위해 일반적으로 제공되고 쉽게 이용할 수 있는 그래픽 객체이다.

<그림 2>는 표준정규분포 함수의 그래프를 6개의 보간점을 이용하여 그린 경우이다. 6개의 보간점을 이용했을 때 최대오차가 오차허용도보다 작기 때문에 적절한 보간점이 될 수 있다. 그러나 보간점의 수가 4개이거나 5개인 경우 최대오차가 오차허용도보다 더 크기 때문에 근사를 위한 보간점으로 사용하는 것은 적절치 못하다.



<그림 2> 확률밀도함수의 그래프

참고문헌

[1] Kim, T. W. and Lee, K. W. (2001), Weight Control and Knot Placement for Rational B-spline Interpolation, *Korean Society of Mechanical Engineers International Journal*, Vol. 15, No. 2, 192-198.

[2] Press, H. W., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P. (1992), Numerical Recipes in C, *Cambridge University Press.*

[3] Wegman, E. J. and Carr, D. B. (1993), Statistical Graphics and Visulization, *Handbook of Statistics*, Vol. 9.

[4] Wilkinson, L. (1999), The Grammar of Graphics, *New York: Springer-Verlag.*