빅데이터 분석을 통한 고장함수 추정 및 예측 방안 구축

국방대학교 윤봉규
국방대학교 문성암
국방대학교 최진우
국방대학교 조원영
국방대학교 남광식

요 약

2020년 국방부 주요 추진 과제에는 '총 수명주기 개념을 적용한 국방획득 및 운영관리 강화'가 포함되며, 여기에는 '전 무기체계, 주요장비 수명주기 간 효율적 운영관리개념 장착', '수리부속 수요예측 정확도 향상 및 재고자산 감축을 포함하고 있다. 군의 장비정비정보체계에서는 총 8가지(산술 평균법, 이동평균법 등) 수리부속 예측기법을 제공한다. 이들은 적은 양의 데이터로 예측이 가능하다는 장점이 있으나, 정확도가 낮다는 단점도 있다. 필요로 하는 데이터의 양이 적다는 것은 빅데이터 기술을 활용할 수 없음을 의미한다. 다시 말하면, 군의 예측 기법으로는 2020 국방부 주요추진 과제를 달성하기에는 한계가 있으며, 최신화된 기술 도입에 대한 연구가 필요하다.

이에 본 연구에서는 2020 국방부 주요추진 과제 중 수리부속 예측과 장비의 수명주기 간 효율적 관리의 기초가 될 수 있는 고장함수를 추정할 수 있는 방안을 살펴보았다.

고장함수란 군의 데이터를 분석하여 추정한 총수명주기간 장비의 고장 발생확률에 관한 함수이다. 우리 군은 데이터를 장기간 축적하지 않으므로, 고장함수 추정에 과거의 연구 모델이나 상용화된 기술을 사용하기에는 한계가 있었다. 그러나, 계층형 베이지안 모델은 현재 군이 보유한 적은 양의 데이터로도 비교적 높은 정확도를 확보할 수 있는 예측 모델로, 모수를 확률분포로 가정하며 사용자의 이해가 쉽고 분석이 편리하다. 계층형 베이지안 모델 추정에 사용된 통계언어인 Stan은 많은 계산을 동반하는 모수 적합속도를 비약적으로 향상시켜 복잡한 모델도 비교적 빠르게 추정할 수 있다는 장점이 있다.

단계형 분포는 과거의 이력에 상관없이 현재상태에 의해서 미래가 결정되는 성질을 가진 마코프체인을 기초로 하는 확률모형이다. 과거의 연구에서 확률분포를 추정하기 위해서 특정 분포를 가정하였던 것과 달리, 단계형 분포에서는 특정 분포를 가정하지 않아도 실제와 가까운 확률분포를 추정할 수 있다. 또한, 단계형 분포 추정 과정에는 EM 알고리즘, Moment Matching 등 여러 방법론이 존재하여 다양한 적합을 시도해 볼 수 있다. 이에 본 연구에서는 계층형 베이지안 고장함수 외에 단계형 분포 고장함수를 추가적으로 연구하고, 향후 활용방안을 모색하였다.

본 연구에 활용된 데이터는 해군의 장비정비정보체계에서 확보한 98척에 해당하는 10년(2009~2019년) 분량의 고장 관련 데이터 28,013건이다. 이들 데이터로부터 함정의 총수명은 약 31년임을 추정하였고, 함정별로 수명에 따라 정리한 데이터로부터 해군 함정의 고장 데이터는 함정별, 타입별로 수량이 다르고 분포도 다른, 비균일한 특징이 있음을 확인할 수 있었다. 해군 함정의 구조적인 특징에 따라 3개의 계층으로 구분하고 수명 연차별 고장 데이터의 분포를 확인한 결과, 수명 초반과 말기에 고장이 많고 수명 중반에 고장이 적은 개략적인 욕조모양의 형태가 도출되었다. 데이터를 바탕으

로 계층형 베이지안 모델을 구성하여 고장함수를 도출하였고, 과거에 성능이 입증된 자기회귀누적이동평균법(Autoregressive Integrated Moving Average, 이하 'ARIMA') 및 프로핏(Prophet) 알고리즘과 성능을 비교하여 군에 사용하기에 적합함을 입증하였다.

단계형 분포의 고장함수 추정과정은 2단계로 이루어졌다. 먼저, 수명 연차별 고장건수에 대한 확률분포를 추정하였으며, 추정된 확률분포를 바탕으로 수명 연차별 고장건수 기댓값을 산출하였다. 1단계 수명 연차별 고장건수 확률분포를 추정하는 과정에서 주어진 데이터에 가장 적합한 단계형 분포 추정 방법를 비교 분석하여 데이터의 성격에 따라 사용하는 방법론을 제시하였다. 2단계에서는 수명 연차별 고장건수 기댓값을 반영하여 총수명주기간 고장함수를 추정하는 방법을 제시한다.

연구결과, 계층형 베이지안 모델은 계층 간 정보공유를 통해 군 데이터의 비균일성을 보완하여 비교적 높은 정확도를 보였다. 군의 보안 정책으로 인해 향후 지속될 것으로 판단되는 군 데이터의 비균일성 문제는 계층형 베이지안 모델로 극복할 수 있다고 판단된다. 단계형 분포는 실제와 유사한 확률분포를 도출할 수 있다는 장점을 확인했다. 이에, 계층형 베이지안 모델로 거시적인 관점에서 총 수명간 고장의 형태를 파악하고, 단계형 분포로 미시적인 관점에서 확률분포를 도출한다면 고장함수의 정확도가 증대될 것으로 판단된다.

4차 산업혁명의 기본은 데이터에서 시작한다. 통계 기술이 발달하여 좋은 모델을 적합할 수 있더라도, 데이터가 없다면 큰 한계에 봉착할 수밖에 없다. 일반적인 통계 모델과 같이 본 연구의 고장함수 모델도 데이터가 많을수록 더 나은 모델을 추정할 수 있다. 이는 고장함수에만 국한되지 않으며 모든 통계분석의 공통적인 사항이다. 그러므로 군은 4차 산업혁명에 적합한 과학적인 분석과 운영을 위해 데이터를 체계적으로 관리하고 축적해야한다. 군은 이러한 기능을 전담하기 위한 데이터웨어하우스(Data Warehouse) 구축을 고려해야 한다. 본 연구에서는 활용도 높은 데이터웨어하우스 구축을 위해 반영해야 할 내용을 제시했다.

Ⅰ. 서 론

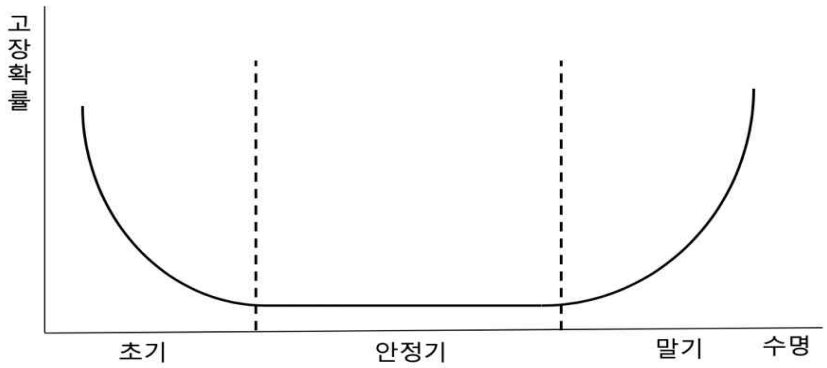
1. 연구배경, 필요성 및 기대효과

2020년 국방부 주요추진 과제 핵심목표 중 하나는 ‘스마트 국방 혁신 강군’이다. 세부목표에는 4차 산업혁명 기술 적용이 목표로 포함되었다. 상기 목표의 세부 전략 3가지 중 하나로 ‘총 수명주기 개념을 적용한 국방획득 및 운영관리 강화’가 포함되며, 이는 ‘전 무기체계, 주요장비 수명주기 간 효율적 운영관리개념 장착’, ‘수리부속 수요예측 정확도 향상 및 재고자산 감축(머신러닝, 딥러닝 기법 활용, 소요산정 방법 등 개발 및 적용)’을 주요 내용으로 다루고 있다. 수명주기 효율적 운영관리와 수요예측 정확도 향상을 위해서는 장비의 특성(고장 특성)을 파악하는 것이 중요하다.

한편, 군은 장비정비정보체계를 이용하여 수리부속의 수량을 예측하고 있다. 장비정비정보체계에서 사용할 수 있는 수리부속 예측 방법은 총 8가지(산술평균법, 이동평균법 등)가 있다. 이 방법들은 적은 양의 데이터로 예측이 가능하다는 장점이 있으나 정확도가 낮다는 단점이 있다. 필요로 하는 데이터의 양이 적다는 것은 4차 산업혁명의 빅데이터 기술을 활용할 수 없음을 의미한다.

민간기업들과 학계에서는 컴퓨팅 능력의 향상에 따라 빅데이터를 활용하고 있다. 관련 논문들이 많이 쏟아져 나오고 있으며, 빅데이터를 활용한 고장 예측방법도 다양하게 개발되고 있다. 이런 방식으로 예측한 고장 특성을 민간기업에서는 실무에 활용하여 비용을 절감하고 운영 효율성을 높이고 있다. 예를 들어, 휴대폰은 제작 후 바로 고객에게 판매되지 않는다. 제작 초반에 발생하는 다량의 고장들을 정비한 후에 고장이 감소하는 안정화 상태에 돌입하면 고객에게 판매된다. 고객이 구매한 휴대폰은 고장이 거의 없이 몇 년간 사용된 후 고장이 잦아진다. 학계에서는 일반적으로 총수명주기 간 고장확률의 분포는 <그림 1>과 같은 욕조형태를 따른다고 알려져 있다.

휴대폰이 고객에게 판매되는 시점은 초기와 안정기의 구분지점이 된다. 휴대폰의 수명이 다하여 고장이 잦아지는 곳은 말기에 해당한다. 초기와 말기에는 고장이 많으므로 정비를 위한 수리부속을 많이 준비해야 한다. 그러나 현재 군의 예측방법들은 이와 같은 수명주기에 따른 고장확률의 변화를 고려하지 않는다. 따라서, 군에서 고장 특성을 고려한 고장함수를 개발하여 실무에 적용한다면 국방개혁의 주 목표인 국방 운영 효율성을 제고할 수 있을 것이다.



<그림 1> 욕조형태의 고장확률분포

고장함수 활용의 구체적 효과를 살펴보면 다음과 같다. 정확도 높은 고장함수가 군의 장비에 적용되면 매년 발생 가능한 고장의 횟수를 예측할 수 있다. 그리고 장비의 정비 정책 결정과 정비 예산의 감축에 활용할 수 있다. 고장 확률이 적은 장비를 실전에 우선 배치하는 전략을 세우는 등의 작전 운용 정책 결정에도 활용할 수 있다. 또한, 고장 횟수를 통해 수리부속의 필요량을 도출할 수 있다. 현재 군에서 적정량의 수리부속을 구매하는 예산이 연 평균 1.21조원인 것을 고려하면, 예산 절감에 큰 효과가 있을 것으로 기대된다. 고장함수를 이용하여 예측정확도를 1% 향상시키면 약 121억원의 수리부속 구매예산이 절감된다고 추정할 수 있다.

2. 연구목표에 따른 연구범위와 방법

앞 절에서 설명한 연구 목적 달성을 위해 본 연구의 연구목표와 연구방법을 살펴보면 다음과 같다.

가. 고장함수 도출을 위한 빅데이터 수집 및 관리방안 제시

본 연구에서는 해군 장비정비정보체계에서 수집한 함정들의 추진용 엔진 고장 데이터들을 분석하였다. 이 과정에서 군사자료는 오래 보관되지 않고 고장함수 추정에 필요한 정보가 충분하지 못함을 파악했다. 또한 현재 군에는 민간 수준의 빅데이터 수집체계가 구축되어 있지 않음을 확인하였다. 이에, 향후 고장함수 추정의 정확도를 향상시키기 위한 군 데이터 활용방안을 제시한다.

나. 국내·외 민간에서 활용하고 있는 예측기법 활용 동향 분석 및 군 적용 방법론 제시

고장함수와 관련된 연구 문헌을 조사하고 현재 사용 중인 예측 기법의 분석을 통해 우리 군 적용 가능성을 판단한다. 본 연구에서는 최근에 작성된 논문들을 중심으로 예측 기법을 조사하고, 군의 데이터 특성을 고려하여 군에 적용 가능 여부를 판단한다. 이를 토대로 군의 고장함수 추정/예측에 적합한 방법론을 제시한다.

다. 실무 활용이 가능한 고장예측 방안 제시(분석 알고리즘 세부 라이브러리와 코드 제시)

수집된 빅데이터를 활용하여 실제 군에 적용할 수 있는 모델을 프로그래밍하는 과정과 함께 소스코드를 제공하여, 향후 군에 실제 적용 시 참고할 수 있는 예시를 제공한다.

Ⅱ. 고장함수 추정 방법

군에서 활용중인 예측기법은 총 8가지(산술평균법, 이동평균법, 가중이동평균법, 최소자승법, 단순지수평활법, 이중지수평활법, 홀트지수평활법, 윈터지수평활법)이다. 8가지 예측기법은 다양해보이지만, 큰 범주에서 구분하면 산술평균, 이동평균, 지수평활법으로 구분된다. 이들 기법은 적은 양의 데이터를 활용할 때 사용되는 기법이므로 빅데이터 시대에 발전한 컴퓨팅 파워와 분석 기법에 비해 분석결과의 수준이나 적용 분야의 범위 측면에서 상대적으로 한계가 있다. 또한, 과거에는 보안상의 문제로 기초자료의 수집이 어려워 정확도가 낮은 예측 모델을 유지할 수 밖에 없었다. 본 연구에서는 이런 문제를 개선하기 위해 과거에 군에서 사용했던 기법과 더불어 빅데이터 분석기법의 하나인 베이지안 통계와 단계형 분포를 활용하는 방법을 살펴본다.

빅데이터 이전의 전통적인 통계학의 예측기법은 ARIMA, 지수평활(Exponential Smoothing), 주기 및 추세 분할(Seasonal, Trend Decomposition) 등 여러가지 형태로 개발되어 왔다(Hyndan & Athanasopoulo, 2018). 전통적인 통계학에서는 확률분포의 모수를 단일의 상수로 단정지어 생각해왔다. 반면, 베이지안 통계는 모수 자체를 하나의 확률분포로 두며, 사용자의 직관과 같은 추가정보를 활용하여 데이터가 불충분한 경우에도 현실 적합성이 높은 분석이 가능한 방법을 제공한다. 그리고 단계형 분포는 복잡한 현실을 그대로 표현할 수 있는 장점이 있다. 두 방법은 컴퓨팅 파워의 향상으로 비교적 최근에 예측 모델 개발에 많이 활용되고 있는 방법이며 전통적인 예측/분석 기법을 보완할 수 있다.

1. 고장함수 추정 관련 기존 연구 및 사례

일반적으로 고장함수는 앞서 제시한 <그림 1>과 같은 욕조모양 형태를

가진다고 알려져 있다. Sherbrooke(2006)은 욕조모양 고장 분포 추정을 위해 포아송 확률분포(Poisson Distribution)를 이용한 통계적 분석의 알고리즘으로 컨스트럭티브 알고리즘(Constructive Algorithm)을 제안하였다. 이 알고리즘은 파레토 최적화(Pareto Optimality)를 기반으로 하였으나, 알고리즘의 모수를 추정하는 과정이 복잡하고 까다로워 실무에서 활용가능한 수준으로 일반화하기에는 한계가 있다. Zammori et al.(2020)는 Sherbrooke(2006)의 모수추정 문제를 시간의 흐름에 따라 변하는 와이블 확률분포(Weibull Distribution)의 모수 추정을 통해 해결하고자 하였다. 그러나 컨스트럭티브 알고리즘과 같이 모수 추정의 어려움으로 일반화에 한계가 있음을 스스로 밝힌 바 있다. 군 데이터는 굉장히 많은 종류가 있기 때문에, 일반화가 어려운 모델에 적용하면 데이터 수집/관리가 복잡하여 활용에 한계가 있다.

Wang & Yin(2019)은 와이블 확률분포를 이용하여 욕조모양의 형태를 가지는 고장함수의 초기, 안정기, 말기를 구분짓는 지점을 추정하여 각 구간에 해당하는 와이블 확률분포의 모수를 추정하였다. 도출된 와이블 분포를 추세요소로 설정하고, 실제 데이터를 바탕으로 확률적 요소를 ARIMA 기법으로 예측하였다. 두 추정값을 종합하여 총수명주기간 고장함수를 도출하였다.

이 외에도 수명주기간 고장의 분포를 파악하기 위한 연구가 많이 이루어졌다. Dikis & Lazakis(2019)는 장비의 고장센서의 신호들을 측정하고 각각의 신호들이 실제 고장으로 연결될 확률을 베이지안 네트워크를 활용하여 시계열 예측을 하였다. 이 방법을 활용하기 위해서는 우선적으로 센서의 신호를 전달할 수 있는 장비와 시설이 필요하지만, 설치 비용이 상당하다. 그러나 이는 미래에 우리 군이 나아갈 방향이 될 수 있다. 세부적인 센서 데이터는 고장의 원인을 보다 정확하게 파악할 수 있게 해준다. 이는 머신러닝, 딥러닝으로 이어지는 과정이 되며, 미래에 AI를 활용한 군 장비관리의 기초가 될 수 있다.

Yoo et al.(2019)는 해군함정의 디젤엔진 정비경향 연구시 AHP(Analytic Hierarchy Process)를 활용하여 고장의 형태와 유사성을 분석하였다. AHP와

코사인 유사도(Cosine Similarity) 분석결과, 해군함정의 디젤엔진은 엔진 사용시간보다 엔진의 도입년도가 유사할수록 정비경향이 유사하다는 것이 입증되었다.

Taylor & Letham(2018)이 개발한 세계적인 SNS기업 Facebook의 프로핏 알고리즘은 시계열 푸리에 분할(Fourier Decomposition) 방법으로 데이터에서 연간, 월간, 주간 등 시계열 주기를 도출한다. 이 주기 요소들을 베이지안 가법(Bayesian Generalized Additive Model)으로 다시 합함으로써 예측을 수행하며, 높은 수준의 정확도를 보인다. 프로핏을 이용한 고장 예측의 사례는 아직 찾아보기 힘들지만 향후 활용 가능성은 높을 것으로 판단된다.

2. 상용화된 예측 기법 민간 활용 사례

국내 대기업들은 기업 나름의 예측 기법을 사용하는 것으로 알려져 있으나 내용과 기법은 기업 비밀사항으로, 확인하는데 한계가 있다. 또한, 알고리즘을 확인한다고 하여도 각 기업별 특화된 알고리즘(기업 특화 파라미터 적용)이어서 완전한 해석에 어려움이 있다. 이에, 군에서 기업의 알고리즘만을 구매하여 활용하는 방식은 한계가 있을 것으로 추정된다. 그럼에도 불구하고 민간기업의 예측 알고리즘을 군에 적용할 수 있는 가능성과 시사점을 파악하기 위하여 조사를 수행했으며, 조사를 통해 2개 업체의 사례를 간접적으로 확인하였다. 이들 기업은 외국계 모기업의 예측 알고리즘을 사용하는 것으로 확인되었다. 본 연구에서는 정보 제공자의 요청에 따라 기업명과 상세 알고리즘은 공개하지 않고 향후 연구 활용을 위해 개요만 제시한다.

패스트푸드 OO기업은 본사에서 제공하는 알고리즘을 온라인 상에 설치하고, 패스트푸드 판매 데이터를 실시간으로 누적한다. 일별, 주별, 월별로 예측을 하여 매장에 예측결과를 제공한다. 패스트푸드의 판매량이 주기성이 강하다는 특징이 있으므로 해당 알고리즘은 프로핏과 비슷한 형태의 시계

열 알고리즘을 사용하는 것으로 판단된다. 시계열을 바탕으로 예측을 수행한다는 점에서 군에 적용하기에 적합하다. 고장량의 변화나, 수리부속 사용량의 변화를 수명에 따른 시계열 데이터로 구성하면 계절에 따른 수리부속 소모량 변화나 고장량 변화 주기를 확인할 수 있다. 단, 군에 즉시 적용하는 것은 불가능할 것으로 판단된다. 데이터의 주기를 파악하기 위해서는 많은 양의 데이터가 필요한데, 군 데이터는 보안 정책상 장기간 보존되지 않기 때문이다. 민간의 데이터 축적과 같이 군에서도 이와 같은 문제를 수정하여 다량의 데이터를 누적할 수 있다면 주기성을 파악할 수 있게 된다. 주기성을 가진 데이터를 이용하면 군에서도 높은 예측 정확도를 가진 예측 모델을 적용할 수 있을 것으로 판단된다.

조선업계 OO기업은 외국계 본사에서 예측 알고리즘을 구매하여 운용한다. 이 알고리즘은 장비마다 센서를 설치하여 과거의 경보 상황 데이터를 누적하여 학습한다. 장비의 부하조건이나 주변 조건 등 운영 환경에 따라 예측 모델이 변경되는 형태의 알고리즘이며, 예측값과 실제 운용상의 실측값에 차이가 발생하면 경보를 주도록 설치된다. 그러나 해당 알고리즘은 장비의 타입이나 연식 등에 구애받지 않으며 센서 설치를 기본조건으로 한다는 단점이 있다. 장비 연식을 고려하지 않는다는 특징은 노후로 인한 고장 특징을 고려하지 않는다는 의미이다. 여기에는 알고리즘 자체를 주기적으로 수정해야하는 한계가 있을 수 있다. 또한, 센서 설치를 기본으로 하므로 구형 기계 장비와 같은 경우에는 설치가 제한될 수 있고, 센서 자체의 고장도 염려해야 한다는 한계가 있다. 고장함수가 일반적으로 욕조형태를 가진다는 점을 고려하면, 이 방법은 장기 계획을 세우기 위한 고장함수를 도출하는데 적합하지 않다고 판단된다.

상용화된 알고리즘을 이용하여 고장함수를 추정하기 위해서는 공통적으로 다량의 데이터를 필요로 한다. 또한, 주기와 추세를 도출하려면 추정할 주기에 맞는 데이터가 필요하다. 일반적으로 통계학에서는 최소 30개 이상의 데이터가 있을 때 데이터의 분포가 신뢰할 수 있는 수준이라고 판단한다. 월 단위 주기를 알기 위해서는 최소 30개월 이상의 데이터가 필요하고, 연

단위 주기를 알기 위해서는 최소 30년 이상의 데이터가 필요하다. 장비의 센서에서 수집되는 신호 데이터의 경우에는 최소 30회 이상의 경보 데이터가 있어야 해당 경보에 대한 추정을 할 수 있다. 3장에서 자세히 설명하겠지만, 군은 데이터를 장기간 축적하지 않는다. 때문에 상용화된 알고리즘의 적용은 현재 상태에서는 한계가 있다. 이에, 상용화된 알고리즘에 관한 연구에서는 군의 데이터 축적 정책에 대한 고찰이 선행되어야 한다.

3. 베이지안 추정법

본 절에서는 연구에서 활용한 계층형 베이지안 고장함수 추정에 관한 통계 기법과 과정을 설명한다. 베이지안 통계는 사건이 일어날 확률을 수치로 단정하지 않고, 분포로 설명한다. 확률을 분포로써 설명하는 이유는 다음과 같다. 예를 들어, 동전 던지기에 대해서 일반적으로 앞, 뒤가 나올 확률은 각각 0.5라고 알고 있다. 그러나 실제로 동전던지기를 10번 한다면 앞, 뒷면이 5번씩 나올 수도 있지만 앞면만 10번이 나올 수도 있다. 왜냐하면 던지는 상황에 여러 영향 요소들이 추가될 수 있기 때문이다. 모든 영향 요소를 수리적 확률로 표현할 수도 있겠지만, 이는 문제를 복잡하게 만들어 일반적인 해를 산출하기 어렵게 만든다. 베이지안 통계는 실제의 상황을 바탕으로 확률을 추정한다. 베이지안 통계에서는 다른 요소들의 영향보다는 관심있는 사건에만 집중한다. 때문에 실제 동전 던지기를 한 결과 데이터를 바탕으로 확률을 추정한다. 앞면이 9번, 뒷면이 1번 나왔다면 앞면이 나올 확률을 0.9로 추정하므로 베이지안 추정은 실증적인 분포를 추정한다고 할 수 있다.

$$p(\theta \mid D) = \frac{p(D \mid \theta) * p(\theta)}{p(D)} \qquad \text{<식 1>}$$

베이지안 통계의 기본 수식은 <식 1>과 같다. D는 데이터(Data)를 의미

하며, 증거(Evidence)라고도 한다. θ는 모수(Parameter)를 의미한다. 베이지안 통계에서는 확률을 분포로 표현하기 때문에 모수의 개념이 포함된다. 모수란, 분포의 모양을 결정하는 일종의 계수이며, 모수의 모양에 따라 확률분포가 결정된다고 할 수 있다. $p(\theta \mid D)$는 사후분포(Posterior)라고 하며, 구하고자 하는 대상이 된다. 데이터 D가 주어졌을 때 고장이 발생할 확률이다. $p(\theta)$는 사전분포(Prior)라고 한다. 베이지안 분석을 수행하기 전, 고장이 나타날 확률을 모를 때 개략적으로 고장분포의 형태를 가정하는 확률분포이다. 함정의 고장 데이터는 함정 고장의 모든 경우를 대변하지 못한다. 때문에 확보한 데이터는 고장이라는 전체 모집단의 표본이다. $p(D)$는 모집단에서 데이터가 샘플로 나올 확률이다. $p(D \mid \theta)$는 가능도(Likelihood)라고 하며, 모수가 데이터를 얼마나 설명하는지에 대해 판단하는 요소이다. 정리하면, 본 연구의 베이지안 고장함수 추정 과정은 데이터 D를 확보하여 고장분포의 형태를 개략적으로 판단($p(\theta)$ 선정)한 후, 고장함수 모델을 구축하여 고장함수($p(\theta \mid D)$)를 추정하는 순서로 진행된다.

베이지안 추정과정의 가장 큰 장점은 현실을 반영하는데 있다. 앞서 동전던지기 예시의 '반드시 0.5의 확률로 앞면이 나오지는 않는다'라는 것처럼, 베이지안 추정은 기존의 지식이나 수학적인 계산이 아닌 데이터에 의존한다. 베이지안 추론에서는 데이터 외에도 전문가의 직관이나 경험적 판단을 적용할 수 있다. 사전분포는 어떤 형태일 것이라고 모델을 세우기 전에 가정하는 분포이다. 이는 알고자하는 분포인 사후분포와 형태가 같지만은 않다. 즉, 사전분포는 '이런 분포이다'라는 것에 대한 가정이다. 사전분포는 모델을 구축하는 사용자가 과거의 경험이나 전문가들의 의견을 통해 정할 수 있다. 데이터의 양이 많아서 분포의 형태를 정확히 알 수 있는 경우가 아니라면, 경험적인 요소가 사전분포를 결정하는데 도움이 될 수 있다. 예를 들면, 과거로부터 고장의 확률은 욕조 모양의 형태를 가진다고 알려져 있다. 욕조 모양의 형태라는 것은 수식으로 정립되지 않고 개략적인 형태만 알려져 있음을 의미한다. 과거의 연구들이 고장의 분포를 욕조모양에 적합(Fitting)하기 위해 와이블이나 포아송 분포를 사용한 예시들을 보면, 경험적

인 요소가 모델 구축에 어떤 영향을 미치는지 알 수 있다. 본 연구의 고장 함수 구축에도 사전분포 선정을 위해 일반적으로 널리 사용되는 정규분포를 가정하였다.

고장함수를 추정하기 위해서는 가장 먼저 분석 대상 데이터가 있어야 한다. 본 연구에는 해군 함정의 엔진 고장 데이터를 적용한다. 고장 데이터의 특징에 대해서는 3장에서 자세히 설명한다.

해군의 고장 데이터는 베이지안 통계 기법 중 계층형 베이지안(Hierarchical Bayesian) 모델에 적합하였다. 계층형 베이지안이란, <식 1>의 사전분포의 모수(θ)를 상수로 두지 않고 분포로 적합함으로써, 모수를 추정하기 위한 베이지안 기본 수식이 추가되는 것이다. 다시 말하면, 단층형 베이지안에서는 모수가 상수로 결정됨으로 사전분포($p(\theta)$)가 결정되었지만, 계층형 베이지안에서는 $p(\theta)$를 설명하기 위해 θ에 대한 분포($p(\zeta)$)가 추가되어 $\theta = p(p(\zeta))$ 형태의 사전분포가 적용된다. $p(p(\zeta))$는 베이지안 기본 수식이 2중으로 들어간 형태이므로 2개의 계층을 가진다고 할 수 있다. 본 연구에서는 해군의 데이터를 3개 계층으로 구성하였다.

계층이 3개가 되는 모델의 수식은 복잡하다. 계층형 모델의 경우 사전분포 중 1개는 반드시 모수를 사용자가 정해야 한다. <식 2>을 보면 그 이유를 알 수 있다.

$$\begin{aligned} \theta_1 &\sim Normal(\theta_2,\ 1) \\ \theta_2 &\sim Normal(\theta_3,\ 1) \\ \theta_3 &\sim Noraml(\alpha,\ 1) \\ \alpha &\sim Normal(\mu,\ 1) \end{aligned} \qquad \text{<식 2>}$$

<식 2>는 3개 계층을 가진 계층형 베이지안 모델의 모수를 간단히 표현한 것이다. 각 θ의 아래첨자로 붙은 숫자는 계층형 모델의 몇 번째 층에 해당하는 모수인지를 표현한 것이다. 1층의 모수 θ_1은 정규분포의 형태를 가지며, 이때 정규분포의 모수는 평균 θ_2와 표준편차 1이다. 마찬가지로 3층에

해당하는 θ_3까지 표현 가능하다. 여기서, θ_3의 모수 α의 분포는 사용자가 모수까지 직접 지정해야 한다. 고장함수의 경우, θ_3는 가장 상위층의 모수가 된다. 3장에서 자세히 설명하겠지만, 계층형 모델의 상위층은 해군 엔진 전체의 수명 연차별 고장 평균값에 해당한다. 모수 μ는 상위층의 수명 연차별 값이 된다. 예를 들어, 계층형 모델에서 1년차에 해당하는 고장의 평균이 10이라면, 사용자는 μ의 값을 10으로 지정할 수 있다. 수명이 30년인 데이터를 3개 계층을 가진 계층형 모델로 추정하기 위해서는 <식 2>과 같은 모델이 30개 필요하게 된다.

계층형 베이지안 모델을 구축하는데 또 하나의 중요한 과정은 사전분포가 어떤 형태의 분포 모양을 가지는지 결정하는 것이다. <식 2>에서는 모든 모수가 정규분포(Normal Distribution)로 가정되었다. 모델의 사전분포를 알기 힘든 경우 일반적으로 정규분포를 가정한다. 이는 베이지안 추정에 관한 과거의 연구에서 경험적으로 얻어진 것으로, 정확한 가정이라고는 할 수 없다. 베이지안 추정에서는 사전분포의 형태에 따라 모델의 결과가 바뀐다. 따라서, 고품질의 데이터가 충분히 존재한다면 정규분포로 가정하는 것보다는 해당 데이터를 분석하여 도출된 분포를 사용하는 것이 더 정확하다.

계층형 구조의 수식을 구축한 후에는 MCMC(Markov Chain Monte Carlo) Sampling 과정을 수행한다. MCMC Sampling은 사전분포에서 나타날 수 있는 임의의 수치들을 반복적으로 추출하고 베이지안 추정 과정을 통해 사후분포를 도출하는 과정을 반복한다. 도출한 사후분포들의 가능도(Likelihood)를 비교하여, 가장 높은 가능도의 사후분포를 최종 모델로 선택한다. 가능도를 이용하여 최종 모델을 결정하는 방법을 최대 가능도 추정(MLE : Maximum Likelihood Estimation)이라고 한다. MCMC 과정에는 Gibbs Sampling, Metropolis- Hastings, Hamilton Markov Chain 등 많은 통계 기법이 포함된다. 단, 통계기술의 발달로 MCMC Sampling을 프로그래밍에 내장된 함수로 간단히 계산가능하기 때문에 추정 과정에 대한 이해가 없어도 사용 가능하다. 따라서 세부적인 통계기법에 대한 설명은 생략한다. 본 연구에서는 통계언어 Stan에서 제공하는 MCMC 함수를 활용하였다.

4. 단계형 분포를 활용한 추정법

단계형 분포(Phase-type Distribution)에 대한 이해는 마코프체인에서 시작된다. 마코프체인은 과거 이력과 현재 주어진 상황 하에서, 미래는 과거 이력과 관계없이 현재에만 의존하는 성질(Memoryless Property)을 가진 확률모형이다. 이 중에서 흡수마코프체인은 여러 일시상태를 거쳐 흡수상태로 전이되면 이후 전이를 종료하는 모형이다. 연속시간 흡수마코프체인의 전이율행렬은 다음과 같은 형식으로 나타낼 수 있다.

$$Q = \begin{bmatrix} T & t \\ \mathbf{0} & 0 \end{bmatrix} \qquad \text{<식 3>}$$

여기서, $\mathbf{0}$은 크기가 $(1\times p)$이고 모든 원소가 0인 행벡터이고, 일시상태에서 흡수상태로 전이되는 흡수율 t는 $t = (t_1, t_2, \cdots, t_p)'$ 인 열벡터이다. 일시상태간 전이행렬 T는 $T = [T_{ij}]$인 $(p\times p)$행렬이다. 이때 크기가$(p\times 1)$이고 모든 원소가 1인 벡터를 e라고 하면 연속형 흡수마코프체인에서 행의 합은 0이 되어야 하기 때문에 $t = -Te$가 된다. $\pi = (\pi_1, \pi_2, \cdots, \pi_p)$인 행벡터는 초기상태확률로 마코프체인의 초기 상태를 나타내는 확률이다. 단계형 분포는 흡수마코프체인에서 초기상태확률 π와 일시상태 간 전이행렬 T를 모수로 가지며, $PH(\pi, T)$로 표현된다. 이는 초기상태확률 π와 일시상태 간 전이행렬 T를 통해 흡수마코프체인을 구성할 수 있기 때문이다. 흡수마코프체인에서 초기확률에 따라 처음 상태에서 흡수상태로 전이될 때까지 걸린 시간을 확률밀도함수로 나타낸 것이 단계형 분포의 확률밀도함수이며, 다음의 <식 4>와 같이 정의된다. 흡수마코프체인과 단계형 분포에 대한 상세한 내용은 윤봉규(2008)를 참조하기 바란다.

$$f(x) = \pi \cdot \exp(Tx) \cdot a \qquad \text{<식 4>}$$

단계형 분포의 모수를 추정하는 방법은 크게 Moment Matching 방법과 EM(Expectation Maximization) 방법이 있다. 본 연구에서 두 가지 방법을 모두 적용하였으며, EM방법은 Asumussen et.al.(1996)이 제시한 EM알고리즘을 사용하였다. 본 연구는 R언어와 Julia언어를 사용하였으며, 세부 코드는 부록 4에 수록하였다.

이산형 자료의 경우에는 추정과정에서 데이터를 간격(Interval)과 비중(Weight)의 형태로 변환하여 Sample Data를 만든 후 단계형 분포를 추정하면 동일한 형태의 확률밀도함수를 추정할 수 있다. 따라서 단계형 분포는 이산형 데이터와 연속형 데이터 모두 확률밀도함수로 표현할 수 있는 장점이 있다.

EM 알고리즘은 최우추정법(Maximum Likelihood Estimation, MLE)으로 결측 자료(Missing Data), 삭제된 자료(Censored Data) 등 불완전한 자료가 가지고 있는 정보를 이용하여 완전한 자료를 도출, 모수를 추정하는 알고리즘적 추정 방법(Algorithmic Estimation)이다(최승석·윤봉규,2010). EM 알고리즘은 기댓값을 구하는 단계(Expectation)와 기댓값을 최대화(Maximization)하는 단계를 반복한다. EM 알고리즘은 주어진 데이터 y가 연속형 데이터인 경우, 이를 연속형 흡수마코프체인에서 초기상태에서 일시상태를 거쳐 흡수상태로 흡수될 때까지 걸린 시간으로 본다. 여기서 흡수마코프체인의 초기확률(π), 일시상태 전이율행렬(T)을 알면 흡수 시까지 걸린 시간의 확률밀도함수를 알 수 있다. 따라서 처음 임의의 초기확률(π), 일시상태 전이율행렬(T)을 설정하여 그에 따른 기댓값을 구하는 과정인 E-step과, 기댓값에 따라 가능도를 최대화하는 초기확률(π), 일시상태 전이율행렬(T)을 구하는 과정인 M-step을 반복한다. 이후 y에 적합도를 높일 수 있는 모수의 변화가 더 이상 발생하지 않으면 반복 과정을 중단하고 단계형 분포의 모수 초기확률(π)과 일시상태 전이율행렬(T)을 도출한다.

단계형 분포는 무기억 속성(Memoryless Property)을 가지는 지수분포를 기본으로 한다. 여기에 사용되는 지수분포는 평균에 의해서 분포의 형태가 결정되므로 현실세계의 다양한 현상을 묘사하지 못하는 단점이 있다. 이를

보완하기 위해 Neuts(1989)는 여러 지수분포의 합으로 이루어진 형태의 분포를 제안하였다. 얼랑(Erlang)분포와 유사한 단계형 분포는 지수분포의 무기억 속성을 유지하면서, 광범위한 확률적 현상을 표현할 수 있다(최승석·윤봉규,2010). 이러한 특성에 따른 고장함수 추정은 장비의 고장건수와 시간에 관한 분포를 비교적 간단한 방법으로 추정할 수 있게 하여, 고장함수 추정 연구에 많이 활용되었다. 각 군의 장비는 특성에 따라 계획정비 주기를 결정하여 운용된다. 계획정비 이후 고장이 발생하는 시간을 측정할 수 있고, 이들의 데이터를 이용하면 단계형 분포로 고장함수를 적합시킬 수 있다. 이와 비슷하게 단계형 분포를 고장함수에 적용한 사례가 있었다.

Faddy(1995)는 석탄 분쇄장비(Coal Pulverising)를 대상으로 고장발생을 흡수상태로 설정하고, 고장이 발생할 때까지 걸린 시간을 단계형 분포로 적합(Fitting)하였다. 이로써 고장 발생시간에 대한 고장함수를 추정하였고, 이는 다양한 형태의 분포를 단계형 분포로 적합할 수 있다는 예시가 되었다. 또한, 단계형 분포의 적합과정에서 단계수를 증가시키고 감소시킴에 따라 적합의 효과를 결정할 수 있다는 특성이 확인되었다. 이러한 특성은 다양한 형태로 존재하는 군 데이터에 존재하는 이상치들을 판단하고 제거하는데 활용될 수 있다.

여러 가지 상황변수들이 적용되는 작전 상황에서는 일반적이지 않은 고장이 발생할 수 있다. 이런 현상들은 장비를 대표하는 고장함수 추정에 저해요소로 작용한다. 따라서 이상 현상이 발생한 데이터들을 이상치로 두고 제거하는 데이터 정제 작업은 반드시 필요하다. 단계형 분포에서 단계를 결정하는 작업은 데이터 정제와 더불어 고장함수의 정합성 수준을 결정할 수 있는 기준이 된다.

Kim & Kim(2017)은 고장 이후 복구가 불가능한 시스템의 대체품 대수 결정을 위해 단계형 분포를 활용하였다. 여기서 시스템을 구성하는 서브 장치들은 서로 다르다는 특징을 가지는데, 장치간의 상태전이 형태를 직렬(Series), 병렬(Parallel) 시스템으로 구분하여 각각에 대한 고장함수를 도출하였다. 군 장비는 대부분 고장 발생시에도 정격 성능을 유지하기 위해 서

브 시스템을 갖추고 있다. 예를 들어, 전원변환장치는 컨버터 고장시 장비의 전원이 Shut-down 되어 장비 구동이 중지되는 위험한 상황이 발생할 수 있다. 서브 시스템은 이런 상황에서 정격성능을 유지하기 위한 장치이다. 단계형 분포의 구현 과정인 상태전이 행렬(Transition Matrix)에도 이와 같은 시스템의 구조를 반영할 수 있다. 단계형 분포는 장비의 서브 시스템들이 모두 중지될 시간을 추적하므로 모든 상황(장비 구조)을 고려한 고장함수를 추정할 수 있다는 장점이 있다.

Barde et al.(2020)은 Power Transformer 장비의 고장 데이터를 분석하여 고장함수를 산출하였다. 삭제된 자료(Censored or Truncated Data)가 있는 불완전한 관측자료(Incomplete Observation)의 고장분포의 모수를 추정하기 위하여 2단계의 추정과정을 거쳤다. 1단계에서 기존에 정립된 고장분포의 형태를 가정하고, 2단계에서는 1단계의 결과를 활용하여 연속시간 단계형 분포로 추정하는 과정을 거친다.

2단계 모수 추정과정은 모수 추정의 정확도를 향상시킨다. 앞서 언급한 바와 같이 군은 보안 정책으로 인해 군 데이터는 일반적으로 불완전한 형태를 가진다. 따라서 위와 같이 2단계에 걸친 단계형 분포 모수 추정은 보존되지 않은 데이터로 인한 고장함수의 정확도 저하를 방지할 수 있는 방법이 될 수 있다. 군 데이터의 특징은 3장에서 자세히 설명한다.

한편, 해군 함정의 고장/정비체계를 단계형 분포로 분석한 사례로, 고재우·김각규·윤봉규(2013)의 연구가 있다. 해당 연구는 해군 함정의 예약정비 시스템의 총비용을 최소화하기 위한 최적의 예약정비 간격을 산출하였다. 해군 함정의 정비관련 시간자료 30여 개를 수집하여 정비단계를 3단계로 나누고, 이를 단계형 분포로 적합하였다. 이때, 지수분포로 적합한 확률밀도함수와 적합도를 비교하여 단계형 분포가 현실 설명력이 우수하고, 응용가능성이 높음을 입증하였다. 이는 본 연구에서 다루고자 하는 해군 함정 고장데이터의 분포 추정 방법으로 단계형 분포를 적용할 수 있는 근거가 되었다.

본 연구에서는 이상에서 설명한 추정기법의 장·단점 비교를 수행했다. 비

교모델로는 ARIMA, 프로핏이 적절하다고 판단하였다. ARIMA는 이동평균법과 자기회귀를 이용하여 높은 정확도를 보이는 수학적 접근방법으로, 현재까지도 경제, 산업 등의 분야에서 활발하게 활용되는 예측기법이며, 고장확률 추정에 사용된 기록도 존재한다. 프로핏은 시계열 분석에 높은 정확도를 보이는 2018년도에 개발된 알고리즘이므로 최신형 알고리즘이라고 할 수 있다. 추가적으로 단계형 분포를 이용하여 고장함수를 추정했지만, 이는 고장발생 시간의 분포를 비교적 간단하게 추정할 수 있는 방법론이다. 하지만 현재 데이터의 비균일적 특징으로 인해 함정 타입별, 함정별 비교가 불가하여 비교모델에서는 제외하였다. 또한, 현재 군에서 사용 중인 이동평균법과 지수평활법은 빅데이터를 활용하지 않는 방법론이므로 비교모델에서 제외하였다. 단, 이들 모델에 해당하는 코드를 부록 1(이동평균 고장함수, 지수평활 고장함수 코드)에 수록하여 관련 연구에 활용될 가능성을 열어두었다.

Ⅲ. 고장함수 추정 및 발전방안

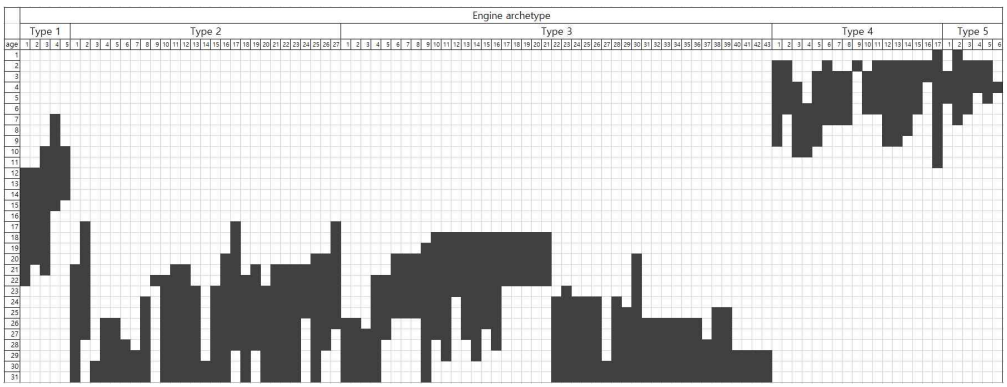
1. 고장함수 추정을 위한 기초 Data 분석

고장함수 추정 대상으로 해군 수상 전투함 추진용 디젤엔진의 고장 데이터를 확보하였다. 총 120여척의 2009~2019년 정비데이터 61,455건을 확보하였다. 정비데이터에는 표준정비, 비표준정비, 항해 중 긴급고장 정비, 정박 중 고장정비가 포함되어 있다. 표준정비는 고장과 무관하게 계획정비 수행 시 실시하는 정비이므로 고장으로 볼 수 없다. 비표준정비는 표준정비 외에 계획정비 시 수행하는 정비이다. 여기에는 성능저하 부품에 대한 정비나 항해 중 발생하였으나 잔고장에 해당하여 현장 정비로 작전 수행이 가능했던 고장건 등이 포함된다. 비표준 정비는 실제 고장과 고장 직전의 상태(성능저하)가 모두 포함되어 있어 고장정비로 분류하였다. 여기에 항해 중 긴급고장 정비와 정박 중 고장 정비 건수를 합하고 대상 함정들 중 예기치 못

한 사고가 발생했던 함정과 데이터 수가 너무 적은 함정은 제외하였다. 최종적으로 총 98척의 함정의 28,013건의 고장 데이터를 분석 대상으로 결정하였다.

함정별 데이터는 수명에 따라 다시 분류했다. 수명에 따른 분류 결과 해군 함정들의 수명은 일반적으로 31년인 것으로 확인되었다. 확보된 데이터를 총 수명 31년으로 두고 각 수명주기에 각 함정별로 데이터가 존재하는 구간을 <그림 2>와 같이 도식화하였다. 함정별로 2009~2019년의 수명 위치를 확인하면 2000년에 도입된 함정은 수명 10~20년에 위치한 데이터를 확인할 수 있다. 한편, 1985년에 도입된 함정은 25~35년의 수명에 해당하는 데이터가 있어야 하지만, 노후된 함정들은 평균적으로 31년 이상의 데이터가 존재하지 않았다. 즉, 해군 함정의 수명을 약 31년으로 추정할 수 있다.

데이터에 포함된 함정들은 함정 엔진의 종류에 따라 5가지 타입으로 분류할 수 있었다. 예를 들어, DDH급 함정에서 운용하는 엔진과 FF급 함정에서 운용하는 엔진은 서로 다른 엔진이므로 엔진 타입에 따라 데이터를 분류를 하였다. 분류된 함정들은 타입별 함정의 수가 모두 달랐다. Type 1의 경우 5척에 불과했고 Type 3은 43척이었다. Type별로 함정이 건조된 시기가 모두 다르기 때문에 데이터를 타입별, 수명 연차별로 정렬하면 데이터의 특성을 쉽게 파악할 수 있다.



<그림 2> 수명 연차별, 함정별 데이터 존재 구간

<그림 2>에서 보듯 총 수명주기 31년에 비해 데이터가 존재하는 구간은 매우 적다. 함정의 타입별로 도입년도가 비슷하여 데이터가 존재하는 위치도 다르고 척수도 다르다. 이런 경우의 통계분석은 쉽지 않다. 수명 연차별로 존재하는 데이터의 수가 상이하고, 데이터의 수가 적은 부분이 많다. 또한, 데이터에는 다량의 노이즈(정보체계 사용자의 오기, 누락 등)가 포함되어 있을 것으로 추측할 수 있으므로 데이터의 신뢰성 자체가 높다고 할 수 없다. 이와 같은 데이터의 비균일성은 모든 군의 공통적인 문제일 것으로 추측된다. 군은 보안상의 문제로 과거의 데이터를 장기간 보존하지 않기 때문에, 파기된 데이터로 인해 비균일성이 나타나게 된다. 그러나 과거의 데이터를 보존하지 않으면 분석에 한계가 생길 수 있고, 분석 자체가 불가능할 수 있다. 군은 과거의 데이터를 지속적으로 누적할 필요가 있다.

2. 베이지안 추정법을 활용한 고장함수 추정

가. 베이지안 추정을 위한 Data 특성 분석

해군의 엔진 데이터에 포함된 정보량은 비균일하다. 비균일한 정보를 가진 데이터를 유사한 구조로 표현할 수 있는 경우 계층형 베이지안 모델을 적용할 수 있다. 이 모델을 적용하면 계층간 정보 공유(Information Pooling)의 특성을 활용할 수 있다(Gelman et al., 2005). 데이터의 특징에 따라 계층을 구분할 수 있다면 계층간의 정보가 풀링되어 정확도 높은 모델을 구축할 수 있다(Gelman et al, 2013; Taieb et al., 2017). 계층형 모델에서는 데이터가 변경되거나 추가되는 경우 모델을 쉽게 업데이트 할 수 있다. 상위층을 포함한 전체의 모델은 초모수(Hyper-parameter)라는 정보로 저장된다. 계층형 베이지안 모델에서 일부 데이터의 변경시 초모수는 수정되지 않고, 해당 부분의 모수(Parameter)만이 업데이트 된다(Gelman, 2006). 업데이트가 용이하다는 특징은 고장함수를 구축한 후에 추가적인 데이터를 지속

적으로 쉽게 적용할 수 있다는 의미가 된다. 다시 말하면, 신규, 퇴역 전력이 지속적으로 발생하는 군의 경우 계층형 베이지안 모델을 사용하면 초모수를 고정시키고, 하위단의 각 전력에 해당하는 데이터만 등록하고 삭제하는 방식으로 고장함수 모델을 지속 업데이트할 수 있다. 따라서 본 연구의 고장함수를 계층형 베이지안 모델로 추정하였다. '다'항에서 상세히 설명하겠지만, 본 연구에서 추정한 계층형 베이지안 고장함수의 경우에는 신규 데이터를 반영하기 위해 사전분포를 재추정할 필요가 없다. 그래서 기존 모델에 신규 데이터만 입력하면 최신화된 고장함수를 산출할 수 있다. 예를 들어, 1~10년까지 데이터가 존재하는 s1이라는 함정의 11년차 고장함수를 추정한다면, 먼저 1~10년까지의 데이터를 바탕으로 사전분포를 추정하여 고장함수를 산출한다. 그리고 산출된 모델에 추후 확보된 11년차의 데이터를 입력하는 것으로 고장함수를 최신화 할 수 있다. 이 과정에서 고장함수의 형태를 좌우하는 초모수, 모수도 최신화 된다. 이와 유사하게 모델의 수정 절차도 간단하다. 만약 산출된 모델에 9년차 데이터의 오류를 발견한다면, 해당 데이터만 수정하여 개선된 고장함수를 추정할 수 있다. 따라서 본 연구에서 구축된 계층형 베이지안 모델은 수정이 용이한 장점이 있다.

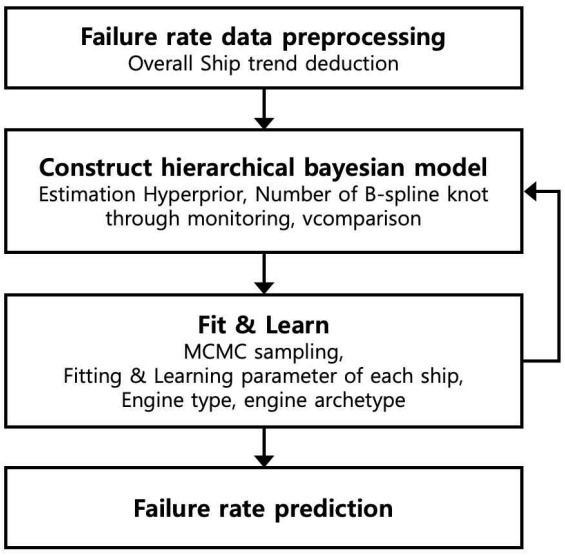
계층형 베이지안 모델에 적용하기 위해 해군 데이터의 비균일적인 특성과 구조적 특성을 확인해야 한다. 비균일성에 관한 특징은 위에서 확인하였다. 구조에 관한 분석 결과 엔진 데이터는 크게 3개 층으로 구분하였다. 3개 층은 각각 ① 엔진들의 공통적인 특성, ② 엔진 타입별 특성, ③ 엔진 고유 특성이다. 예를 들어, s1이란 함정의 엔진은 s1 함정 엔진 고유의 특성을 가진다. 같은 엔진을 쓰는 함정이 모두 같은 고장을 기록하지는 않는다. 모든 조건이 동일하다고 해도 함정의 운용방법에 따른 누적된 피로도나 정비 불완전성의 차이에 따라 고장이 발생하는 정도는 달라질 수 있다. 이해의 편의를 위해 ③, ②, ① 특성 순으로 설명하였다. ③ 엔진 고유의 특성이란, 이와 같이 운용되면서 발생하는 고장의 차이에 해당한다. s1 함정은 t1이라는 함정 타입이다. t1 타입의 엔진은 타입으로 인한 특성을 가진다. s1 함정의 엔진은 함정 타입에 따라 실린더의 수, 출력, RPM 등의 특징이 모두 다

르다. 실린더의 수가 많다면 실린더의 고장수도 많을 것으로 추측할 수 있고, RPM이 높다면 엔진 운용 속도의 차이에 따른 피로의 차이로 고장 빈도가 높거나 낮을 수 있다. ② 엔진 타입별 특성이란, 이와 같이 함정 타입 자체로 인한 엔진의 제원에 따라 발생하는 차이라고 할 수 있다. 확보한 데이터에는 총 5개의 타입이 포함되었다. 이들은 모두 해군 함정의 엔진이라는 공통적인 특징을 가진다. 상선 엔진은 속도의 변화가 거의 없이 장거리를 항해하는데 적합한 엔진이고, 어선 엔진은 자주 시동을 껏다 켜도 바로 작동이 가능해야하는 엔진이다. 해군의 엔진은 시동을 걸고 바로 움직일 수 있으면서 유사시를 대비하여 일정 시간 이상의 고속 운용이 가능해야 하며, 불가피한 시기에는 엔진의 경보 신호를 차단하면서까지 최대 출력을 유지할 수 있는 엔진이어야 한다. 이와 같이 큰 범주에서 나눌 수 있는 엔진의 용도가 ① 엔진의 공통적인 특성이다. 해군 함정의 엔진은 ① ~ ③의 특성을 모두 가진다. 즉, 모든 함정 엔진을 동일하게 3개의 층으로 구성할 수 있다. ㉮ 확보한 함정(98척)의 데이터를 동일한 구조로 표현 가능하다. 또한 ㉯ 확보한 데이터는 수명 연차별 데이터 수가 균등하지 않다는 비균일성을 가진다. 비균일적 특성에 관해 3장에서 서술하였다. 위 계층적 구조와 비균일적 특징을 고려하여 본 연구의 고장함수 모델에 계층형 베이지안 모델을 적용하였다.

나. 고장함수 추정 절차

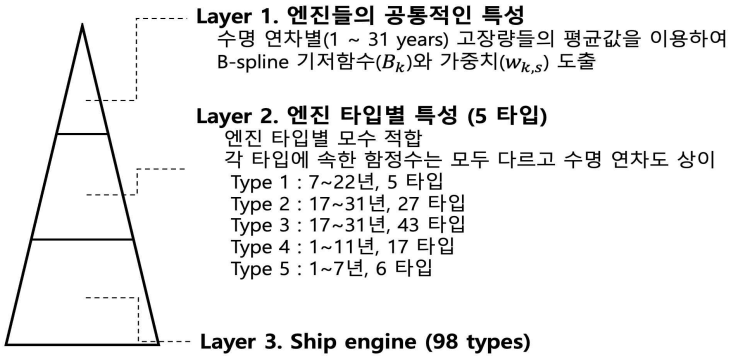
본 절에서는 2.2절의 내용에 따라 고장함수를 추정하는 과정에 대해 기술하였다. 고장함수 추정 순서는 <그림 3>과 같다. 수집된 데이터를 분석하기 위한 형태로 전처리(Failure Rate Data Preprocessing)해야 한다. 수집된 해군의 데이터에서 이상치로 판단되는 값들을 제외하고, 데이터를 수명에 따라 나열하는 과정을 거친다. 수명에 따라 나열하면 해군 함정의 총수명을 추정할 수 있다. 데이터를 엔진의 타입에 따라 분류하여 순서를 나열함으로써 타입별 수명의 위치 분포와 데이터의 개수를 파악할 수 있다. 데이터를

정제하면 개략적인 고장함수의 형태를 추측(Overall Ship Trend Deduction)할 수 있다. 계층형 베이지안 모델을 구축(Construct Bayesian Hierarchical Model)하기 위해서는 먼저, 이산형 데이터인 개략적인 고장함수 형태를 곡선 형태로 변경해야 한다. 본 연구에서는 B-spline Fitting이라는 곡선 추정 방법을 적용한다. 2.3절의 방법으로 계층형 베이지안 모델에 적용되는 사전분포를 결정하고, 모델의 수식을 작성한다. 작성된 수식을 바탕으로 MCMC Sampling을 수행하여 모델의 모수들을 적합하고 최적의 사후분포를 도출한다(Fit & Learn). 추정된 고장함수의 가능도 수준을 확인하여 적절하지 않다고 판단되면, 계층형 베이지안 모델 구축 과정으로 돌아가 사전분포를 수정하는 작업을 반복한다. 최종적으로 추정된 고장함수를 이용하여 총수명주기의 고장율을 예측하고 과거 모델과의 성능을 비교한다.



<그림 3> 고장함수 추정 과정

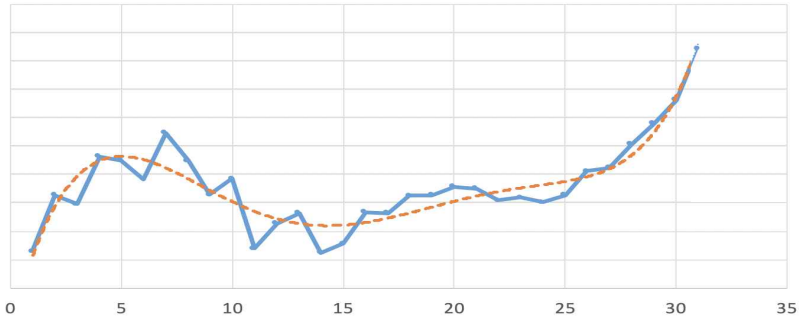
3.1절에서 분석한 데이터의 특징을 고려하면, 3개 계층의 구조를 <그림 4>와 같이 도식화 할 수 있다.



<그림 4> 고장 데이터의 계층형 구조

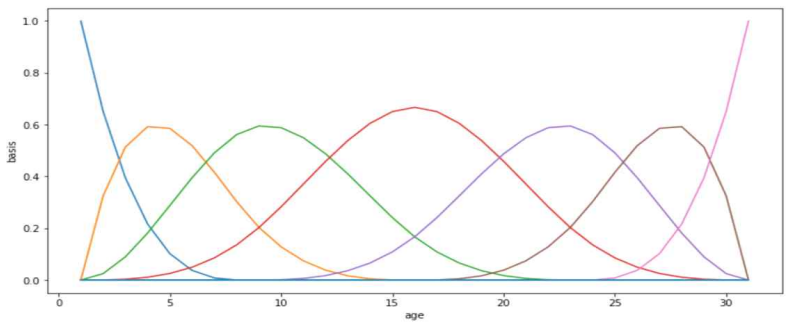
<그림 4>에는 각 층의 데이터와 계층형 베이지안 모델에 적용될 파라미터에 대한 개략적인 설명이 포함되어 있다. Layer 1은 함정 엔진 전체에 관한 층이다. Layer 1에서는 총수명 31년동안 각 수명에 해당하는 평균적인 고장량을 확인해야 한다. 함정의 척수와 무관하게 수명 연차별 고장횟수의 평균값을 산출하여 <그림 5>과 같이 그래프로 도식화 하였다. <그림 5>에서 가로축은 수명을 의미한다. 세로축은 수명에 해당하는 고장횟수이나, 보안상의 문제로 척도는 표기하지 않았다. 실제 분석시에도 고장횟수는 비율조정(Scale)하여 사용하였다. 파랑색 선은 수명 연차별 고장횟수의 변화를 나타낸다. 7년차까지 고장량이 증가추세를 보인 후 감소하였다가, 유지되는 구간을 거친 후 25년차부터 고장량이 급증하였다. 빨강색 선은 추세를 나타낸 것이다. 추세선은 5년까지의 상승구간을 제외하면 학계에서 알려진 바와 같이 욕조모양을 보였다. 1~5년 구간에 고장량이 증가하는 부분은 욕조함수와 다르다. 이 부분은 건조 초기 조선소 하자수리로 군직정비를 실시하지 않은 부분에 해당된다. 확보한 데이터는 군직정비 데이터로, 조선소 하자수리 기록과 건조 시운전 데이터는 전혀 포함되어 있지 않으므로 실제 고장량보다 적게 측정되는 부분이다. 해군은 시운전과 건조 초기 하자수리에 많은 양의 정비를 수행한다. 해당 데이터는 조선소 소유로 확보가 불가능하나, 경험적인 측면에서 보면 초기 부분에 고장량은 상당한 수준이다. 초기 하자수리의

데이터를 확보하여 포함한다면 과거의 연구에서 밝혀진 욕조모양의 형태를 가질 가능성이 클 것으로 판단된다.



<그림 5> 수명 연차별 고장 데이터 분포

고장함수의 형태가 직선은 아니라는 사실을 추측할 수 있으므로 비선형회귀분석을 활용하였다. 본 연구에서는 기저함수 추정법(B-spline Fitting)을 적용하였다. 기저함수 추정법에서는 총 수명을 기저함수교점(knot)이라는 기준에 따라 분할하고 각 부분의 기저함수(Basis Function)와 가중치(Weight)를 도출한다. 여기서 기저함수교점(knot)은 <그림 6>에서 곡선들의 교차점에 해당한다. 기저함수 추정을 적용하면 <그림 6>에서 확인한 수명 연차별 고장 데이터에 대한 기저함수와 가중치를 도출할 수 있다. 기저함수와 가중치는 계층형 베이지안 모델 적합시 곡선형태 고장함수 추정의 기준값이 된다. 이는 다음 절에서 자세히 설명한다.



<그림 6> 고장함수의 기저함수(B-spline)

다. 고장함수 모델 구축

3개 계층의 데이터 구성을 바탕으로 베이지안 계층 구조를 적용하였다. 데이터를 적용할 수 있는 통계 모델을 구축하기 위해서는 모델에 해당하는 파라미터들이 있어야 한다. 전통적인 통계학에서는 파라미터를 상수로 취급하였다. 베이지안 통계에서는 파라미터를 분포로 적용한다. 때문에 파라미터의 구간을 시각적으로 확인하기 용이하고 결과분석도 쉽다. 특히 여러개의 파라미터들이 적용되는 계층형 모델의 경우 파라미터들을 상수로 지정하게 되면 각 층간의 정보 교환이 제한된다. 파라미터를 확률분포의 형태로 적용하면서 각 층간의 정보 교환이 원활해지고 데이터의 적합력도 높아지게 된다. 베이지안 통계에서는 사전분포(Prior)를 사전에 가정하고 사후분포(Posterior)를 추출한다. 사후분포 도출을 위해서는 사전분포 외에도 가능도(Likelihood)와 데이터 실제 분포 정보도 필요하다. 그러나 데이터 실제 분포는 쉽게 계산할 수 없다. 단, 데이터 실제 확률분포는 상수값이며, 결국 사후분포는 사전분포와 가능도의 곱에 비례한다는 점에 기인하여 베이지안 통계에서는 MCMC(Markov Chain Monte Carlo) Sampling 기법을 활용한다.

본 연구에서는 계층형 베이지안 모델 구축에 Andrew Gelman(2012)이 개발한 Stan을 사용하였다. Stan은 베이지안 추정에 최적화된 통계언어이다. 특히, 베이지안 추정의 필수 알고리즘은 NUTS(No-U-Turn Sampler)라는 MCMC Sampling의 일종을 적용한다. 이는 WinBUGS, JAGS와 같은 Sampling 기법에 비해 계산 속도가 굉장히 빠르다. 또한, 모델링 중 발생하는 오류에 대해서도 명확한 오류 사유를 제공하므로 베이지안 추정에 유리하다. Stan에 적용된 본 연구의 계층형 베이지안 모델의 수식은 <식 5>와 같고 이에 대한 Stan 코드는 부록 2에 수록하였다.

Stan 코드를 활용하여 계층형 베이지안 모델을 <식 5>와 같이 구축하였다. $\overline{\alpha_0}$, $\overline{w_0}$ 는 계층 모델의 상위층에 해당하는 파라미터이다. $\overline{\alpha_0}$ 에서 I는 <그림 5>의 수명 연차별 데이터 분포를 선형 회귀로 적합하였을 때 산출되는

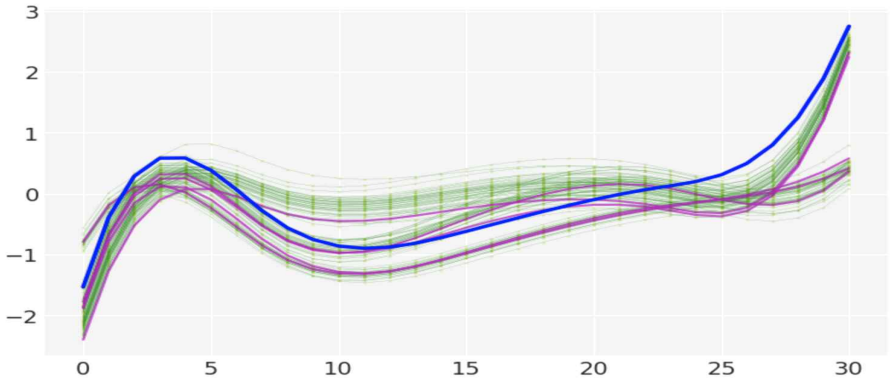
절편(Intercept)이다. $\overline{\alpha_0}$, $\overline{w_0}$ 는 <그림 5>의 수명 연차별 데이터를 B-spline 기저함수에 적합하였을 때 추정되는 사후분포의 평균에 해당한다. 중간층의 파라미터인 $\overline{\alpha_e}$, $\overline{w_e}$ 는 상위층 하이퍼 파라미터를 평균으로 하는 정규분포를 따른다고 가정한다. 그리고 각 배의 고장함수에 해당하는 하위층은 기저함수 파라미터인 $w_{k,s}, B_k$ 와 중간층인 α_s 의 영향을 받는다. 각 층의 파라미터들은 이와 같이 서로 영향을 주고 받게 된다. 이와 같이 계층의 모수들이 서로 연결됨으로써 계층형 모델의 정보 풀링 효과가 나타나게 된다. <식 5>를 부록 2와 같이 Stan 프로그램에 입력하여 MCMC Sampling을 수행하면 고장함수의 사후분포가 도출된다. 수식이 복잡하지만 실제 적용하는 것은 어렵지 않다. <식 5>의 코드는 부록 2에 수록되어 있다. 실무에 계층형 베이지안 모델 적용시 부록 2의 코드를 R 또는 Python에 입력하고 데이터만 업데이트하는 방식으로 지속적인 운용이 가능하다.

$$
\begin{aligned}
Y_s &\sim Normal(\mu_s, \sigma_y) \\
\mu_s &= \alpha_s + \sum_{k=1}^{K} w_{k,s}, B_k \\
\alpha_s &\sim Normal(\overline{\alpha_e}, \sigma_\alpha) \\
w_s &\sim Normal(\overline{w_e}, \sigma_w) \\
\overline{\alpha_e} &\sim Normal(\overline{\alpha_0}, \sigma_{\overline{\alpha}}) \\
\overline{w_e} &\sim Normal(\overline{w_0}, \sigma_{\overline{w}}) \\
\sigma_\alpha &\sim Gamma(10, 10) \\
\sigma_w &\sim Gamma(10, 10) \\
\sigma_{\overline{\alpha}} &\sim Expon
\end{aligned}
\qquad \text{<식 5>}
$$

라. 고장함수 모델 구축 결과 및 성능 확인

계층형 베이지안 고장함수 추정결과는 <그림 7>과 같다. 여기서 파랑색 곡

선은 함정 엔진이 가지는 공통적인 특성에 관한 고장함수이다. 이 고장함수는 기존에 운영하지 않았던 새로운 함형의 함정을 도입할 때 사용할 수 있다. 자주색 곡선은 계층의 중간층에 해당하는 함형 타입별 고장함수에 해당한다. 이는 같은 타입의 함정을 장기간에 걸쳐 계속 건조하는 경우에 사용할 수 있다. 마지막으로 초록색 곡선은 모델 구축에 사용된 각 함정들의 고장함수이다. 이 함수는 해당 함정의 고장량을 예측하기 위해 사용할 수 있다.



<그림 7> 고장함수 도출 결과

각 고장함수의 적용은 상황별 특징을 다음과 같이 반영했다. 첫번째, FF급 함정의 후속 모델로 FFG가 도입되는 경우이다. FFG는 FF를 대체하기 위해 나온 함정이지만 FF 라고할 수는 없다. 탑재된 장비들이나 톤수, 도입년도 등 모두 다르기 때문이다. 그러므로 FFG에는 FF에 해당하는 함형 타입의 고장함수를 적용하면 안되며, FFG를 이전에 운영해본 경험이 없는 신규 전력으로 보아야한다. 이 경우 최상위층의 고장함수를 적용하여, 함정 엔진이라면 가지는 고장의 특성을 반영할 수 있다. 총수명주기간 고장이 얼마나 발생할지 개략적으로 파악할 수 있다. 두 번째, PKG는 2007년경 최초 도입되었고 현재까지 계속 건조중이다. 장기간에 걸쳐 지속적으로 건조하는 함정이므로 해당 타입의 고장함수를 적용할 수 있다. 이미 도입된지 오래된 함정이므로 최상위층의 고장함수를 바탕으로 해당 타입의 고장함수를 도출

가능하다. 운영기간 동안의 데이터와 최상위층으로 도출한 고장함수는 총수명주기의 고장을 예측한다. 즉, 동일한 함형 타입의 함정이 지속 건조되는 경우 과거의 경험으로 산출한 고장함수로 총수명주기간 고장률을 예측할 수 있다. 또, 계층형 베이지안 모델의 초모수와 하위 모수간의 관계를 고려하여, 데이터를 지속적으로 추가하여 고장함수를 정교하게 업데이트 할 수 있다. 마지막으로 PKG 타입의 s1이라는 함정은 2007년에 도입하여 약 15년을 운영하였다. 이 함정의 남은 15년의 고장률을 예측할 때 s1 함정에 해당하는 초록색 고장함수를 활용할 수 있다. 이 함수는 최상위층과 중간층의 초모수들의 영향을 받으면서 최하위층의 데이터에 도출한 모수로 이루어진 고장함수이다. 다시 말하면 해당 함정의 고유한 고장 특징이 최대한 반영된 고장함수이다. s1 함정은 s1 함정 고유의 고장함수를 이용하여 남은 수명기간 중 고장을 예측할 수 있다.

비교모델의 정확도 측정 척도로는 RMSE(Root Mean Square Error)를 사용하였다(Hyndman and Koehler, 2006). RMSE 계산 수식은 <식 6>과 같으며, 오차들을 모두 양수로 치환하여 이들의 평균을 구한 것과 같다.

$$RMSE(\theta_1, \theta_2) = \sqrt{E((\theta_1 - \theta_2)^2)} = \sqrt{\frac{\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2}{n}} \qquad <식\ 6>$$

<식 6>에 따라 계층형 베이지안, ARIMA, 프로핏 모델 및 각 계층별 고장함수의 RMSE를 산출하였으며, 그 결과는 <표 1>과 같다. <표 1>에 제시된 값은 각 계층의 RMSE의 평균값이다. 중위층의 경우 5개의 고장함수가 존재하므로 표의 값은 5개 고장함수 RMSE의 평균값이다. 하위층은 98척 각 함정 RMSE의 평균이다. 각 계층의 정확도 비교결과 계층형 베이지안 모델이 모든 계층에서 RMSE가 가장 낮으므로 성능이 가장 좋다고 할 수 있다. 계층형 모델은 초모수 공유를 통해 정보를 공유하기 때문에 비교모델에 비해 많은 양의 정보를 활용하여 예측을 수행한다고 할 수 있다.

<표 1> 계층별 고장함수 RMSE 비교

계 층	계층형 베이지안	ARIMA	Prophet
상위층	1.0349	1.2587	1.0875
중간층	1.0353	1.0894	1.0741
하위층	1.0274	1.1415	1.1421
평 균	1.0325	1.1632	1.1012

3. 단계형 분포를 활용한 고장함수 추정

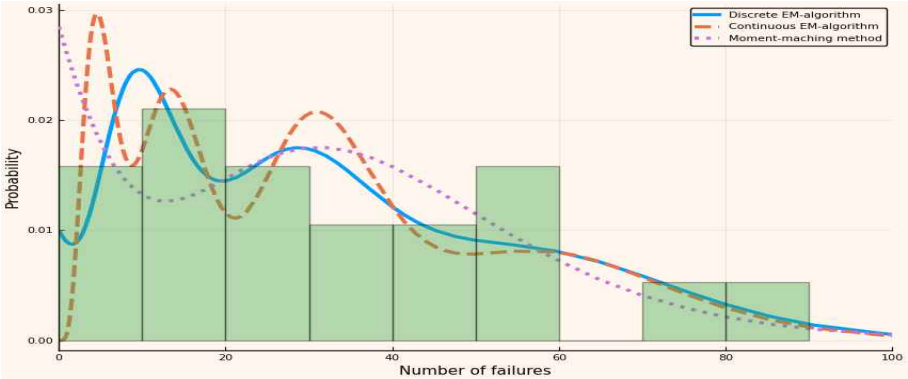
가. 수명 연차별 고장건수에 대한 단계형 분포 추정

본 연구에서는 해군의 전투함 고장분포를 추정하기 위해 2단계에 걸쳐 총수명주기에 대한 고장확률분포를 추정하였다. 단계형 분포는 베이지안 추정법과 달리 데이터가 없는 부분에는 데이터를 생성하여 추정할 수 없다. 따라서 이러한 한계를 극복하기 위해 2번의 분포추정를 실시하였다. 1단계는 세부적으로 2단계의 과정을 거친다. 1-① 각 수명 연차별로 고장건수에 대한 확률분포를 추정하였다. 1-② 수명 연차별 확률분포의 기댓값들을 산출하였다. 이 과정에서 각 수명 데이터에서 부족한 정보를 보완하였다. 2단계는 이 기댓값들을 이용하여 총수명주기(31년)간 고장건수의 분포를 추정하였다.

단계형 분포를 추정하기 위한 방법에는 입력데이터 형태(이산형/연속형)에 따른 EM 알고리즘, Moment-matching 방법을 사용하였다. 계산을 위해 EM 알고리즘은 Julia언어의 EMpht 패키지, Moment-matching은 R언어의 mapfit library를 활용했다. EM 알고리즘은 데이터의 성격에 따라서 연속형일 경우 그대로 데이터를 사용하면 되지만 이산형일 경우에는 간격(Interval)과 비중(Weight)의 형태로 변환하여 샘플 데이터(Sample Data)를

만든 후 사용할 수 있다. 이산형 데이터의 샘플 데이터변환은 Julia언어로 구현한 함수와 EMpht 코드에서 간격 개수(Bin) 설정으로 가능하다. 자세한 방법은 부록 4를 참조바란다. 한편, 단계형 분포 추정을 위해 Julia와 R을 동시에 활용해야 비교나 향후 분석이 용이하다. 따라서 R과 Julia를 연동시켜 EMpht와 mapfit을 연동시켜 활용하는 방법을 부록 3에 제시하였다.

<그림 8>은 단계형으로 추정한 3년차 고장함수로, <식 4>를 활용해서 확률밀도함수로 표현한 결과이다.

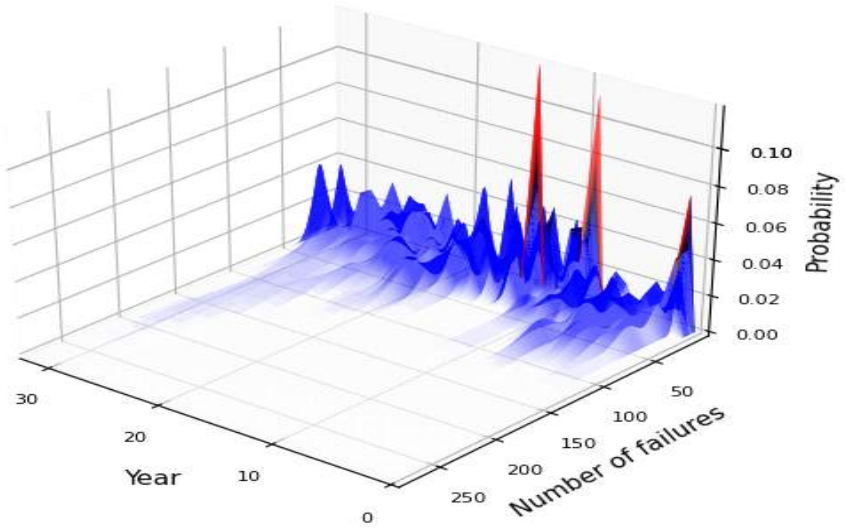


<그림 8> 3년차 함정의 고장건수 분포에 대한 추정결과(1-①단계)

가로축(x축)은 고장발생 건수, 세로축(y축)은 고장 발생확률(밀도)이다.이다. 위에 제시한 3가지 방법(이산/연속형 데이터 입력 EM 알고리즘, Moment-matching 방법)을 모두 사용하였으며 적합도 비교 결과 이산형 데이터 입력 EM 알고리즘이 가장 적합도가 높았다. Moment Matching 적합에서는 고장건수 0~50건 구간에서 시작과 10~20건 구간에서 EM 알고리즘에 비해 낮은 성능을 보였다. 한편, 분포 추정에 사용된 데이터는 이산형 성격을 가진 데이터로 이산형 데이터 입력 EM 알고리즘의 적합도가 연속형 데이터 입력 EM 알고리즘보다 높았다. 특히 데이터의 범위(4～81)가 넓고 개수(19개)가 많지 않은 불완전한 데이터이기 때문에 연속형 데이터 입력 EM 알고리즘 사용시 데이터가 없는 구간에 대해서 과적합이 이루어져서

이산형 데이터 입력 EM 알고리즘에 비해 변화가 심한 형태의 분포를 보여주었다. 따라서 데이터의 성격에 따른 방법의 선택도 중요하지만 데이터의 범위와 개수 등 종합적인 판단을 통해 연속형과 이산형 데이터 입력 EM 알고리즘을 선택하는 것이 중요하다. 그리고 이산형 데이터의 경우에도 샘플 데이터 변환 시 구간(Interval)설정에 대한 고려가 필요하며, 사전에 히스토그램을 통해 샘플 데이터의 적절한 구간 개수(Bin)를 찾아야 한다.

각 수명 연차별 고장건수의 확률밀도함수는 <그림 9>에 제시되어 있다.



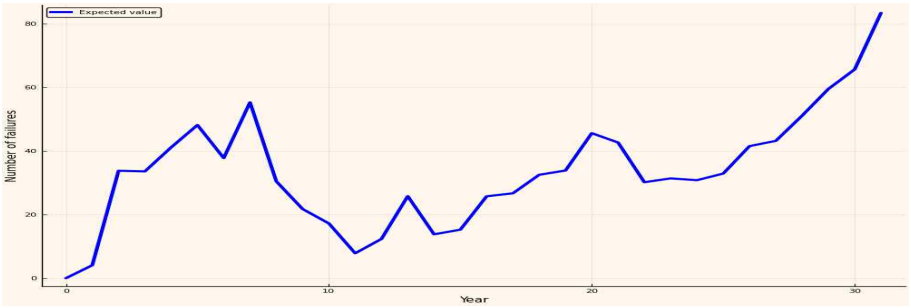
<그림 9> 수명 연차별 고장 확률밀도함수(1-①)

<그림 9>에서 제시한 1-①에서 산출한 수명 연차별 확률분포들에서 고장건수 기댓값을 산출할 수 있다. 기댓값을 산출하는 방법은 확률밀도함수를 활용하거나, 흡수마코프체인에서 흡수시까지 걸린시간을 산출하는 방법을 통해 구할 수 있으며 흡수마코프체인을 이용하는 방법은 <식 7>와 같다.

$$E[x] = \pi \cdot (-T)^{-1} \cdot e \qquad \text{<식 7>}$$

수명 연차별 확률분포의 기댓값은 해당 수명을 대표하는 고장량이 된다. 총

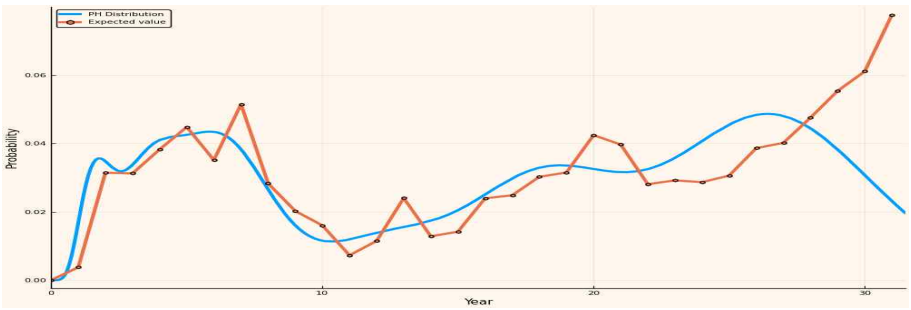
수명주기에 대한 대푯값들의 분포를 <그림 10>와 같이 추정하였다.



<그림 10> 총수명주기간 고장 대푯값의 분포(1-②)

나. 총수명주기간 고장함수 추정

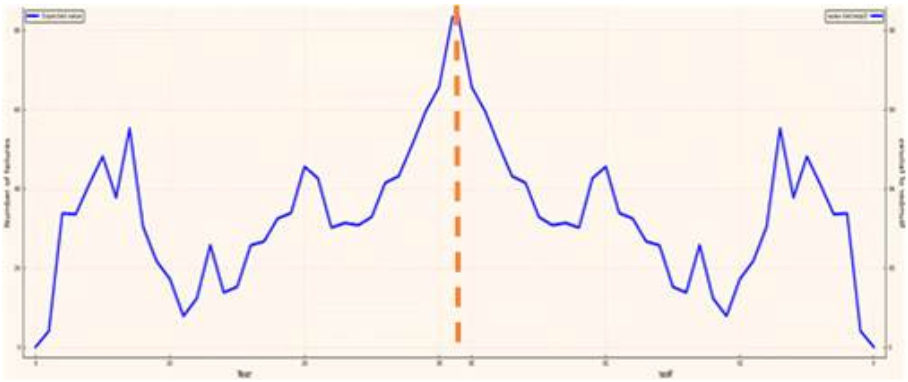
총수명주기간 고장 대푯값의 분포를 이산형 입력데이터 EM 알고리즘으로 <그림 11>와 같이 적합하였다. 총수명주기간 고장분포 추정 결과, 그래프의 초반부는 적합도가 높으나, 중반 이후부터는 적합도가 낮아짐을 시각적으로 확인할 수 있다. 이는 긴꼬리를 가지는 지수분포의 특성에서 기인하는 효과이다.



<그림 11> 고장분포 추정 결과(1-③)

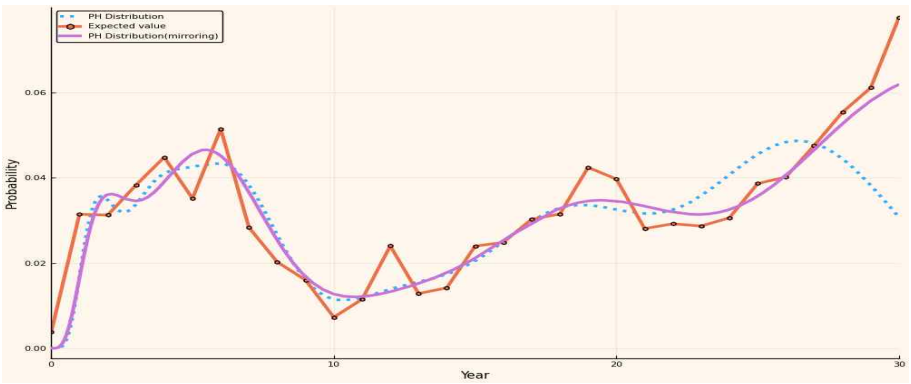
본 연구에서는 지수분포 긴꼬리 특성의 효과를 완화하기 위한 방법을 고안하였으며 이를 미러링(Mirroring) 기법이라 명명하였다. 미러링 기법은 총수명주기간의 기댓값을 31년차의 y축을 기준으로 선대칭시킨다. 미러링을

통해 31년치의 정보는 <그림 12>과 같이 62년치의 정보가 된다.



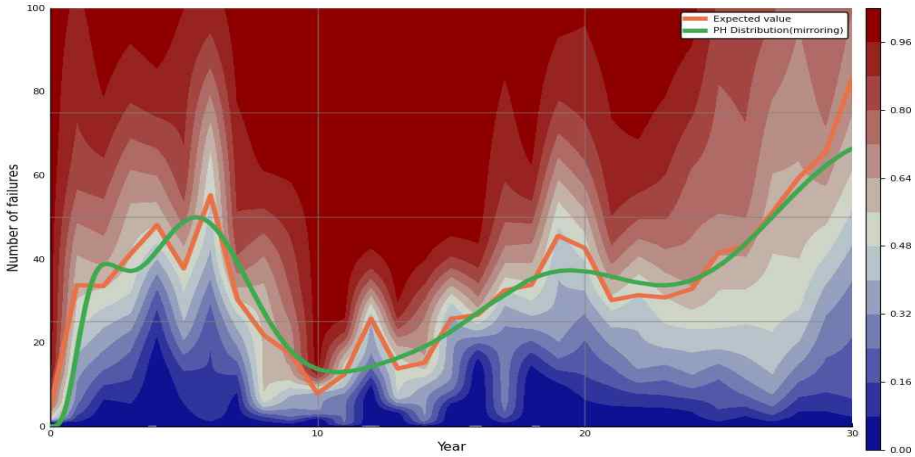
<그림 12> 데이터 미러링

이를 이산형 단계형 분포로 적합한 후 필요한 부분인 1년차에서 31년차까지를 절단하는 방법이다. 미러링 기법을 적용한 결과는 <그림 13>와 같다.



<그림 13> 미러링 기법을 적용하여 도출한 고장함수

<그림 13>의 파랑색선과 자주색선은 미러링 기법 적용 전후의 고장함수이다. 긴꼬리 특성 효과로 인해 적합하지 못하던 수명 후반 부분에 대해 보완된 모습을 확인하였다. 추정한 고장함수를 누적분포함수(CDF)로 변환하고, 수명, 고장건수의 데이터를 <그림 14>와 같이 도식화하였다.



<그림 14> 총수명주기간 고장건수에 대한 누적분포함수

x축은 수명, y축은 고장건수, 고장발생 확률(CDF)은 Color Bar로 표현하였다. 수명 연차별 누적확률이 0.5가 되는 지점과 각 수명 연차별 고장건수 기댓값, 고장함수가 비슷한 형태로 표현되었다. 3가지 확률들은 모두 전반적으로 욕조함수(Bathtub)의 형태를 가진 것을 확인할 수 있다. 수명 초반부터 중반까지는 비교적 분포의 밀도가 높게 나타났다. 반면, 중반 이후부터는 고장건수의 분포(분산)가 점점 넓어짐을 알 수 있다. 즉, 해군 함정은 수명이 오래될수록 고장이 많이 발생하며, 고장예측에 대한 불확실성도 같이 높아진다고 할 수 있다.

베이지안 추정법에서는 불완전한 데이터를 계층으로 나누어 모수를 공유하면서 부족한 데이터를 보충하였다. 따라서 전체 해군 함정의 고장함수와 함정 타입별, 함정 개별 고장함수를 추정할 수 있다. 그러나 단계형 분포는 데이터가 없는 부분에 데이터를 생산할 수 없기 때문에 함정의 타입별, 함정별 고장함수를 추정하는데 한계가 있다. 그러나 <식 4>와 같이 명시적인 형태의 고장함수를 제시하여 추가적인 분석이나 고장 특성 비교에 활용할 수 있는 장점이 있다. 따라서 Data의 불균형을 고려해서 베이지안 추정법만 활용하기 보다는 현 수준에서는 단계형 분포를 보완적으로 활용

하고 향후 Data 수집이나 축적을 통해 단계형 분포를 활용할 수 있는 체계를 만드는 것이 필요하다.

4. 고장함수 정확도 향상을 위한 제언(군 데이터 활용 방안)

해군 함정의 엔진고장 데이터를 분석하면서 분석의 가장 큰 문제점은 비균일성과 데이터의 신뢰도였다. 이러한 문제점을 해결하기 위해 베이지안 모델과 단계형 확률분포를 특성을 활용하였다. 두 방법론의 장·단점은 <표 2>에 정리하였다. 결론적으로 베이지안 모형은 비균일적인 불완전한 데이터를 활용하여 전문가 의견을 반영한 고장함수를 추정할 수 있는 기법이기는 하지만 명시적 함수 형태의 분포로 나타낼 수가 없고 사전분포 가정에 따라서 편차가 크며 연산시간이 소요된다는 단점이 있다. 단계형 확률분포 추정법은 이를 보완할 수 있다. 그러나 단계형 분포는 과적합 발생 가능성과 비균일적 데이터에 대한 한계를 보였다. 본 연구에서는 이산형 데이터 입력 EM 알고리즘을 통해 데이터 양의 부족을 해결하기 위한 방안을 제시하기는 했지만 데이터가 존재하지 않는 수명연차에 대해서 이 방법을 확장하기에는 근본적인 문제가 있었다. 따라서 고장함수 추정의 정확도 개선을 위해서는 데이터 수집 단계에서 활용 가능성을 고려하는 것이 필요하며 이에 대해 본 연구에서는 다음과 같은 데이터 축적/수집 관련 개선점을 제안한다.

본 연구에서 활용한 계층형 베이지안 모델은 비균일성 데이터에 적합한 방법이었다. 정보 풀링의 특성은 부족한 데이터를 대신할 수 있는 일부 정보를 추가로 제공하는 것과 같다. 단, 이는 데이터가 적은 경우보다 많은 경우에 더 큰 효과를 보인다. 해군의 데이터 중 특히 말기에는 함정별 고장발생에 대한 편차가 컸다. 데이터가 충분히 있다면 고장함수가 보이는 편차들을 줄일 수 있고 정확도 높은 모델을 구축할 수 있다.

<표 2> 베이지안 모델과 단계형 확률분포 비교

구 분	베이지안 모델	단계형 확률분포
장 점	• 비균일 데이터 보완 가능 • 전문가의 직관 반영 가능	• 명시적 분포 함수로 표현 • 사전분포 가정 필요없음 • 데이터 특성 반영 용이
단 점	• 명시적 분포로 표현 불가 • 사전분포에 따른 편차 발생 • 연산시간 소요	• 과적합 발생 가능 • 데이터 비균일시 활용 한계

한편 적은 양의 데이터는 신뢰도 측면에서 문제를 야기할 수 있다. 베이지안 모델의 경우 데이터 양이 적을 경우 사전분포를 정규분포로 가정해야 하는 한계가 있으며, 단계형 확률분포는 사전분포가 필요 없기는 하지만 주어진 데이터가 많을수록 현실 적합도가 높아지는 것은 부정할 수 없다. 데이터의 양이 충분하다면 베이지안 모델에서 정규분포가 아닌 실제 고장발생 분포를 적용할 수 있고, 단계형 확률분포에서는 더 정확도 높은 확률분포를 추정할 수 있다. 이런 측면에서 일정 수준 이상의 데이터를 확보하는 것은 고장함수 추정의 기본 요소이다.

데이터를 체계적으로 누적 관리하여 균일한 데이터를 확보하기 위해서는 데이터 웨어하우스를 운영하는 것이 바람직하다. 각 군의 데이터는 국방부 산하의 연구기관에서 데이터 웨어하우스를 통해 체계적으로 관리할 필요가 있다. 한편, 연구기관은 데이터의 누적뿐만 아니라 불필요한 데이터와 필요한 데이터를 구분할 수 있는 능력이 있어야 한다. 군에서 데이터를 수집한다면 수많은 데이터가 수집되고, 종류도 다양하며, 중복되는 데이터도 많을 것이다. 이들 데이터 중 불량데이터를 구분하고 체계적으로 관리하기 위해서는 데이터 분석 전문가가 상존할 수 있는 기관이 필요하다. 예를 들어 본 연구에서 활용된 고장 데이터는 장비정비정보체계의 데이터인데 오기, 누락 등으로 인한 비현실적인 불량 데이터가 상당한 수 포함되어 있었다. 이런 경우, 데이터의 불량여부 식별은 해당 군의 실무 경험자만 식별이 가능하다. 데이터 웨어하우스는 일반적인 사무 부서의 형태가 되어서는 한계가 있

으며 각 군의 실무자와 데이터 분석가가 근무해야 하며, 분석된 데이터를 이용하여 군에 활용할 수 있는 방안을 찾는 정책적인 제안을 할 수 있는 연구인원이 포함되어야 한다.

데이터 신뢰도 저하의 원인은 체계의 접근이 제한되고, 공개용 데이터의 양이 매우 부족하다는 것이다. 이는 체계를 통해 얻어지는 데이터를 활용하고, 공유하고, 또다른 정보가 생성되는데 큰 걸림돌로 여겨진다. 따라서 사용자들은 체계를 통한 정보획득보다는 관련 주관부서의 생성된 자료에 더 많은 신뢰를 보이고 있다. 그리고 아직까지도 수기작성과 체계사용을 병행하고 있다. 이는 체계 사용자 중에서 데이터를 제공하고 저장하는 방법의 접근성이 수기작성에 비해 편의성이 떨어지기 때문이다. 이는 곧 체계에서 데이터 제공자들의 적극적인 참여를 이끌어 낼 수 없으며, 결국엔 체계 내 데이터의 신뢰도에 큰 영향을 미칠 수 있다. 이러한 상황이 지속되면 관련 체계는 사용자들에게 외면을 받을 수 밖에 없다. 따라서 높은 데이터의 신뢰성을 확보하기 위해서는 체계 사용자들의 적극적인 활용을 이끌어내야 한다. 특히 체계를 통해 모두가 쉽게 데이터에 접근할 수 있고 제공받을 수 있다면, 그 활용범위가 늘어날 것이다. 또한 체계 사용자들의 데이터 활용도와 충성도가 높아지고, 각종 데이터 기록에 적극적으로 참여하는 선순환 구조를 만들 것으로 기대된다. 여기에 데이터가 빠짐없이 축적될 수 있도록 시스템을 구축하는 것이 중요하다.

마지막으로 빅데이터는 스마트폰 위치정보나 카드사 고객의 구매정보와 같이 애초에 분석 목적으로 수집한 데이터가 아니라 활동에 자연스럽게 녹아 있는 데이터이다. 이런 특징으로 인해서 빅데이터는 데이터 수집비용이 거의 발생하지 않으며, 그 결과 대규모 데이터를 적은 비용으로 수집할 수 있다. 이런 맥락에서 빅데이터의 핵심은 저비용의 대용량 데이터를 분석해서 가치를 찾는 과정이라고 할 수 있다. 군에서 빅데이터 활용을 통해 가치를 증대시키기 위해서는 빅데이터의 저비용 데이터 활용이라는 핵심 가치에 대해서 초점을 맞출 필요가 있다. 즉, 빅데이터 분석을 위해서 추가적인 데이터 수집 절차를 만들어 고비용의 데이터를 축적하기 보다는 업무 절차

에서 자연스럽게 데이터가 축적되도록 하고, 이를 이용하려는 발상의 전환이 필요하다. 고장함수 추정을 위해서도 업무 절차에 자연스럽게 녹아서 축적되는 데이터를 찾고 이를 분석하는 프로세스 정착시킨다면 많은 양의 저비용 데이터(Cheap Data)에서 높은 수준의 가치(High value)를 창출할 수 있을 것이다.

Ⅳ. 결론

본 연구에서는 해군 수상 전투함(98척)의 추진용 디젤엔진의 고장함수를 추정하였다. 고장함수를 추정하는 과정은 기본 데이터의 특성을 분석하고, 이를 최적으로 적합할 수 있는 추정법을 모색 및 적용하는 순서를 따랐다. 먼저, 해군 수상 전투함의 추진용 디젤엔진의 고장 데이터를 분석한 결과, 군의 데이터 특성상 장기간의 데이터가 축적되어 있지 않고, 정보체계 전환 시 이전 정보체계의 데이터 손실로 비균일한 특성을 가지고 있었다. 고장 데이터의 비균일한 특성을 고려한 고장함수 추정 방법을 찾기 위해 최근 고장예측 동향과 현재 국내 대기업에서 사용하고 있는 상용화된 예측기술, 고장함수에 관한 문헌연구를 실시하였다. 이를 통해, 계층형 베이지안 모델과 단계형 분포 모델을 고장함수 모델 구축 방법으로 선택하였다. 계층형 베이지안 모델은 계층간 정보 공유 (Information Pooling)의 장점을 가진 모델이다. 특히, 이 모델은 데이터가 특정한 구조를 가지면서 비균일한 특성을 가질 때 적용 가능하다. 반면, 군은 단일 장비를 운용하지 않는다. 예를 들어, 함정을 건조하는데 있어 단일 타입의 함정을 여러 대 건조하는 것과 같다. 공군이 같은 타입의 전투기를 여러 대 동시에 도입하고, 육군이 동일한 전차를 대량 생산하는 것과 같다. 이 때문에 군용 장비들은 개별 장비, 장비의 타입, 전체 장치로 총 3가지 층으로 구조화할 수 있다. 구조화된 데이터는 구조의 각 계층에서 정보가 풀링되어 높은 정확도의 모델을 추정할 수 있게 한다. 계층형 베이지안 고장함수를 추정 하고 모델의 성능 측정을 위해

최신형 알고리즘인 Prophet과 과거로부터 널리 사용되고 고장함수 추정에 활용된바 있는 ARIMA를 비교모델로 지정하였다. RMSE 비교결과, 계층형 베이지안 모델의 RMSE값이 가장 낮아 고장 함수 추정 방법으로 적합함을 입증하였다. 그러나 베이지안 모델은 일반적인 단일함수에 의한 고장함수 산출이 불가하고, 산출 과정이 복잡하기 때문에 확률분포에 대한 사전지식이 필수적이다. 또한, 한번 구축된 모델은 수정이 어렵고, 데이터의 양이 적어도 연산 시간이 길고 명시적 확률분포를 도출하지 못하는 단점이 있다.

본 연구에서는 베이지안 추정법의 한계를 보완하기 위해 단계형 분포를 활용하여 고장함수를 산출하였다. 단계형 분포는 2단계 추정과정을 거쳐 도출된다. 총수명주기(31년)를 1년 단위로 구분하여 수명 연차별 고장분포를 먼저 추정하였다. 추정한 연단위의 고장분포에서 기댓값을 산출하고 수명 연차별 기댓값들의 분포를 단계형 분포로 다시 추정함으로써 총수명주기의 고장함수를 추정하였다. 2단계의 추정 방법을 통해서 수명 연차별 고장분포를 추정할 수 있다. 수명, 고장건수, 고장 확률 3차원을 도식화하여 전체적인 고장의 특성을 쉽게 표현할 수 있다. 단계형 분포는 데이터의 형태에 따라 연속/이산형 데이터 입력 EM 알고리즘을 선택해야 한다. 단계형 분포는 어떤 형태의 데이터로 함수형태의 확률밀도함수로 표현할 수 있으며 확률분포함수(CDF), 기댓값을 모수로 쉽게 산출할 수 있는 장점이 있다. 또한 지수분포와 같은 무기억성(Memoryless Property)을 가지고 있기 때문에 그 활용분야가 매우 넓다. 향후 연구범위를 확장하여 해군 함정과 같은 다양한 체계들의 조합으로 이루어진 시스템을 분석하기 위해 하부 체계들의 고장분포를 3차원(수명, 고장건수, 확률)으로 분석하여 각 체계별 분석결과를 종합하여 상위 시스템의 고장분포를 분석할 수 있다. 단계형 분포는 마코프체인에 대한 사전지식만 있으면, 확률분포에 대한 사전 가정없이 비교적 간단한 코드와 패키지로 빠른 연산이 가능하다. 또한 지수분포의 조합을 통해 복잡한 현실세계를 묘사할 수 있어, 그 활용분야가 매우 넓다. 그러나 지수분포의 조합이기 때문에 극단적인 형태의 데이터나 그래프의 경우 추정이 어려우며, 현재까지 적정한 단계 수 선정기준이 불명확하다는 한계를 가지

므로 두 방법을 상호 보완적으로 활용하는 것이 바람직하다.

<참고문헌>

고재우,김각규,윤봉규 (2013), “예약도착 대기행렬을 활용한 함정정비 최적 예약시간산정에 관한 연구”, 한국경영과학회지, 38(3), 13-22.

윤봉규 (2008), “단계형 확률과정과 국방분야 응용 사례”, 군사과학연구 제1권 제1호, 13-26.

윤봉규 (2020), “마코프이론 및 응용”, 국방대학교, 2020년 2월 16일.

최승석,윤봉규 (2010), “EM 알고리즘을 활용한 단계형 고장/서비스 시간 분포의 모수 추정”, 군사과학연구 제4권 제2호, 25-34.

Abbas, O.M., Mohammed, H.I., Omer,E.A. (2011) “Development of predictive Markov-chain condition-based tractor failure analysis algorithm,” Research Journal of Agriculture and Biological Sciences, Vol.7 No.1, 52-67.

Afenyo, M., Khan, F., Veitch, B., & Yang, M. (2017). “Arctic shipping accident scenario analysis using Bayesian Network approach,” *Ocean Engineering,* 133, 224-230.

Asmussen, S., Olle Nerman, Marita Olsson. (1996) “Fitting Phase-Type Distributions via the EM algorithm,” Scand J Statist Vol 23.

Barragan, J. F., Fontes, C. H., & Embiruçu, M. (2016), “A *wavelet*-based clustering of multivariate time series using a multiscale SPCA approach,” *Computers & Industrial Engineering, 95*, 144-155.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P. & Riddell, A. (2017), “Stan: A probabilistic programming language,” *Journal of Statistical Software 76(1)*.

Chang, Y., Zhang, C., Wu, X., Shi, J., Chen, G., Ye, J., ... & Xue, A. (2019), "A Bayesian Network model for risk analysis of deepwater drilling riser fracture failure," *Ocean Engineering*, 181, 1-12.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990), "STL: a seasonal-trend decomposition. *Journal of official statistics," 6(1)*, 3-73.

Dagum, E. B., & Bianconcini, S. (2016), "*Seasonal adjustment methods and real time trend-cycle estimation,"* Springer International Publishing.

De Gooijer, J. G., & Hyndman, R. J. (2006), "25 years of time series forecasting. *International journal of forecasting," 22(3)*, 443-473.

Dikis, K., & Lazakis, I. (2019), "Dynamic predictive reliability assessment of ship systems," *International Journal of Naval Architecture and Ocean Engineering*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). "*Bayesian data analysis*. Chapman and Hall/CRC."

Gelman, A. (2006), "Multilevel (hierarchical) modeling: what it can and cannot do," *Technometrics, 48(3)*, 432-435.

Heungseob Kim, Pansoo Kim. (2016). "Reliability models for a nonrepairable system with heterogeneous components having a phase-type time-to-failure distribution," Reliability Engineering and System Safety 159, 37-46.

Hyndman, R. J., & Athanasopoulos, G. (2018). "*Forecasting: principles and practice,"* OTexts.

Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G., & Shang, H. L. (2011), "Optimal combination forecasts for hierarchical time series," *Computational Statistics and Data Analysis, 55(9),* 2579-2589.

Hyndman, R. J., & Koehler, A. B. (2006), "Another look at measures of forecast accuracy," *International Journal of Forecasting, 22*, 679-688.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002), "A state space framework for automatic forecasting using exponential smoothing methods," *International Journal of forecasting, 18(3)*, 439-454.

Khanlarzade, N., Yegane, B., Kamalabadi, I., & Farughi, H. (2014), "Inventory control with deteriorating items: A state-of-the-art literature review," *International journal of industrial engineering computations*, *5*(2), 179-198.

Kuo, R. J., & Li, P. S. (2016). "Taiwanese export trade forecasting using firefly algorithm based K-means algorithm and SVR with wavelet transform," *Computers & Industrial Engineering, 99*, 153-161.

Mackey, T. B., Barney, J. B., & Dotson, J. P. (2017). "Corporate diversification and the value of individual firms: A Bayesian approach," *Strategic Management Journal, 38(2),* 322-341.

McElreath, R. (2020). "*Statistical rethinking: A Bayesian course with examples in R and Stan,"* CRC press.

M.J.Faddy.(1995), "Phase-type distributions for failure times," Mathematical and Computer Modeling, Volume 22 Issues 10-12, 63-70

Moon, H-J., Choi, J-W. and Lee, H-S. (2020), Failure prediction in hierarchical equipment system: spline fitting naval ship failure, Stancon 2020(mc-stan.org)

Neuts MF. (1989), "Structured Stochastic Matrices of M/G/1 Type and Their Applications," Marcel Dekker, Inc., New York.

Scheu, M. N., Tremps, L., Smolka, U., Kolios, A., & Brennan, F. (2019). "A systematic Failure Mode Effects and Criticality Analysis for offshore wind turbine systems towards integrated condition based maintenance strategies," *Ocean Engineering*, 176, 118-133.

Sherbrooke, C. C. (2006). "*Optimal inventory modeling of systems: multi-echelon techniques* (Vol. 72)," Springer Science & Business Media.

Shor, B., Bafumi, J., Keele, L., & Park, D. (2007), "A Bayesian multilevel modeling approach to time-series cross-sectional data," *Political Analysis, 15(2)*, 165-181.

Shumway, R. H., & Stoffer, D. S. (2017), "*Time series analysis and its applications: with R examples,"* Springer.

Shumway, R. H., & Stoffer, D. S. (2017), "Time series analysis and its applications: with R examples. Springer.Sterman, John D. "Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment," *Management science* 35.3 (1989): 321-339.

Stephane Barde, Young Myoung Ko, Hayong Shin. (2020). "Fitting discrete phase-type distribution from censored and truncated observations with pre-specified hazard sequence," Operations Research Letters 48, 233-239

Tabandeh, A., & Gardoni, P. (2015), "Empirical Bayes approach for developing hierarchical probabilistic predictive models and its application to the seismic reliability analysis of FRP-retrofitted RC bridges," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering, 1(2)*, 04015002.

Taieb, S. B., Yu, J., Barreto, M. N., & Rajagopal, R. (2017), "Regularization in hierarchical time series forecasting with

application to electricity smart meter data," In *Thirty-First AAAI Conference on Artificial Intelligence.*

Taylor, S. J., & Letham, B. (2018), "Forecasting at scale," *The American Statistician, 72(1)*, 37-45.

Van der Auweraer, S., & Boute, R. (2019), "Forecasting spare part demand using service maintenance information," *International Journal of Production Economics*, *213*, 138-149.

Vehtari, A., & Lampinen, J. (2002), "Bayesian model assessment and comparison using cross-validation predictive densities," *Neural computation, 14(10)*, 2439-2468.

Vehtari, A., Gelman, A., & Gabry, J. (2017), "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC," *Statistics and Computing, 27(5),* 1413-1432.

Wang, J., & Yin, H. (2019), "Failure Rate Prediction Model of Substation Equipment Based on Weibull Distribution and Time Series Analysis," *IEEE Access*, 7, 85298-85309.

Yoo, J. M., Yoon, S. W., & Lee, S. H. (2019), "SNA-based Trend Analysis of Naval Ship Maintenance," *Journal of the Korea Society of Computer and Information*, 24(6), 165-174.

Zammori, F., Bertolini, M., & Mezzogori, D. (2020), "A constructive algorithm to maximize the useful life of a mechanical system subjected to ageing, with non-resuppliable spares parts," *International Journal of Industrial Engineering Computations*, 11(1), 17-34.

Appendix 1. 이동평균/지수평활 고장함수 코드

1. 이동평균 고장함수 코드

* 파이썬(Python) 언어를 활용하며, 3이동평균을 기본으로 구현

가. 파이썬 패키지 불러오기

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
```

나. Data 로딩

```
failure_df = pd.read_csv('failure_count.csv').set_index('age')
u = np.nanmean(failure_df.values.flatten())
s = np.sqrt(np.nanvar(failure_df.values.flatten()))
failure_df = pd.DataFrame((failure_df - u) / s, index=failure_df.index,
columns=failure_df.columns)
```

다. 3이동평균 구현

```
failure_df['avg'] = failure_df.mean(axis=1)
failure_df['MA'] = 0
for i in range(1, failure_df.shape[0]-2):
    if i == 1 :
        failure_df['MA'].loc[1] = failure_df.avg[1]
```

```
    elif i == 2 :
        failure_df['MA'].loc[2] = failure_df.avg[1]
    elif i == 3 :
        failure_df['MA'].loc[3] = failure_df.avg[1] + failure_df.avg[2]
    else :
        failure_df['MA'].loc[i+3]= (failure_df.avg[i] +
failure_df.avg[i+1]+failure_df.avg[i+2]) / 3

plt.figure(figsize=(6,4))
plt.plot(failure_df.avg)
plt.plot(failure_df.MA)
plt.show()
```

2. 지수평활 고장함수 코드

* 파이썬(Python) 언어를 활용하며, 평활상수 $\alpha = 0.7$ 기준으로 구현

가. 파이썬 패키지 삽입

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
```

나. Data 로딩

```
failure_df = pd.read_csv('failure_count.csv').set_index('age')
u = np.nanmean(failure_df.values.flatten())
```

```
s = np.sqrt(np.nanvar(failure_df.values.flatten()))
failure_df = pd.DataFrame((failure_df - u) / s, index=failure_df.index,
columns=failure_df.columns)
```

다. 지수평활 고장함수 구현

```
failure_df['avg'] = failure_df.mean(axis=1)
failure_df['ES'] = 0
alpha = 0.7

for i in range(1,31):
    if i == 1:
        failure_df['ES'].loc[1] = failure_df.avg[1]
    else :
        failure_df['ES'].loc[i] = alpha*failure_df.avg[i-1] +
(1-alpha)*failure_df.ES[i-1]

plt.figure(figsize=(6,4))
plt.plot(failure_df.avg)
plt.plot(failure_df.ES)
plt.show()
```

Appendix 2. 계층형 베이지안 고장함수 코드

* 파이썬(Python) 언어를 활용하며, Stan 코드를 포함

1. 파이썬 패키지 불러오기

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import PowerTransformer
from pystan import StanModel
from scipy.interpolate import splev
from fbprophet import Prophet
import pmdarima as pm
import itertools
```

2. Data 로딩

```
failure_df = pd.read_csv('failure_count.csv').set_index('age')
u = np.nanmean(failure_df.values.flatten())
s = np.sqrt(np.nanvar(failure_df.values.flatten()))
failure_df = pd.DataFrame((failure_df - u) / s, index=failure_df.index,
columns=failure_df.columns)
engine_df = pd.read_csv('engine.csv').set_index('ship')
test_failure_df = pd.read_csv('failure_count_test.csv').set_index('age')
```

```
test_failure_df = pd.DataFrame((test_failure_df - u) / s, index=test_failure_df.index,
columns=test_failure_df.columns)
test_engine_df = pd.read_csv('engine_test.csv').set_index('ship')
```

3. B-spline 기저함수 적합

```
degree = 3
knots = np.linspace(1, 31, num=5)
knots_padded = np.concatenate((np.full(degree, 1), knots,
np.full(degree, 31)))

basis = []
for i in range(knots_padded.shape[0]):
    c = np.zeros(knots_padded.shape[0])
    c[i] = 1
    basis.append(splev(np.arange(1, 32), (knots_padded, c, degree)))
basis_df = pd.DataFrame(basis).transpose().set_index(np.arange(1,
32))
```

4. Stan code 삽입

```
model_code ='''
data {
    int<lower=1> K; // number of knots
    int<lower=1> N; // number of datapoints
    int<lower=1> T; // maximum age
    int<lower=1> S; // number of ships
```

```
  int<lower=1> E; // number of engines
    int<lower=1> age[N];
    int<lower=1> ship[N];
    int<lower=1> engine[S];
    matrix[T,K] B;
    vector[N] Y;
}
parameters {
    // first layer
    real mu_alpha_bar;
    real<lower=0> sigma_alpha_bar;
    vector[K] mu_w_bar;
    real<lower=0> sigma_w_bar;

    // second layer
    real alpha_bar[E];
    real<lower=0> sigma_alpha;
    vector[K] w_bar[E];
    real<lower=0> sigma_w;

    // third layer
    real alpha[S];
    vector[K] w[S];
    real<lower=0> sigma_y;
}

transformed parameters {
    vector[N] mu;
```

```
    for (n in 1:N) {
        mu[n] = alpha[ship[n]] + B[age[n]] * w[ship[n]];
    }
}

model {
    mu_alpha_bar ~ normal(0, 1);
    mu_w_bar ~ normal(0, 1);
    sigma_alpha_bar ~ exponential(1);
    sigma_w_bar ~ exponential(1);

    for (e in 1:E) {
        alpha_bar[e] ~ normal(mu_alpha_bar, sigma_alpha_bar);
        w_bar[e] ~ normal(mu_w_bar, sigma_w_bar);
    }
    sigma_alpha ~ gamma(10,10);
    sigma_w ~ gamma(10,10);

    for (s in 1:S) {
        alpha[s] ~ normal(alpha_bar[engine[s]], sigma_alpha);
        w[s] ~ normal(w_bar[engine[s]], sigma_w);
    }
    sigma_y ~ exponential(1);

    Y ~ normal(mu, sigma_y);
}

generated quantities{
```

```
    vector[N] log_likelihood;
    for (n in 1:N) {
        log_likelihood[n] = normal_lpdf(Y[n]|mu[n], sigma_y);
    }
}
'''
sm = StanModel(model_code=model_code)
```

5. 데이터 적합(Fitting)

```
K = basis_df.shape[1]
T = 31
S = failure_df.shape[1]
E = np.unique(engine_df['engine']).shape[0]
Y = failure_df.values[~failure_df.isnull()]
age_index, ship_index = np.where(~failure_df.isnull())

data = {
    'K' : K,
    'N' : failure_df.values[~failure_df.isnull()].shape[0],
    'T' : T,
    'S' : S,
    'E' : E,
    'age': age_index + 1,
    'ship': ship_index + 1,
    'engine': engine_df.loc[failure_df.columns, 'engine'],
    'Y': Y,
```

```
    'B': basis_df,
}
fit = sm.sampling(data=data)
fit_df = fit.to_dataframe().mean()
samples = fit.extract()
```

6. 고장함수 도식화

```
alpha = np.array([fit_df[f'alpha[{s+1}]'] for s in range(S)])
w = np.array([[fit_df[f'w[{s+1},{k+1}]'] for k in range(K)] for s in range(S)])
mu = np.tile(np.expand_dims(alpha, 0), (T, 1)) + basis_df.values @ w.transpose()

alpha_bar = np.array([fit_df[f'alpha_bar[{e+1}]'] for e in range(E)])
w_bar = np.array([[fit_df[f'w_bar[{e+1},{k+1}]'] for k in range(K)] for e in range(E)])
mu_bar = np.tile(np.expand_dims(alpha_bar, 0), (T, 1)) + basis_df.values @ w_bar.transpose()

mu_alpha_bar = fit_df['mu_alpha_bar']
mu_w_bar = np.array([fit_df[f'mu_w_bar[{k+1}]'] for k in range(K)])
mu_zero = np.repeat(mu_alpha_bar, failure_df.shape[0]) + basis_df.values @ mu_w_bar

fig, ax = plt.subplots(figsize=(10, 6))
ax.set_xlabel('age')
ax.set_ylabel('failures')
```

```
ax.scatter(failure_df.index[age_index], Y, color='m', alpha=0.3)
ax.plot(np.arange(1, T + 1), mu, color='g', alpha=0.3, linewidth=0.9)
ax.plot(np.arange(1, T + 1), mu_bar, color='m', alpha=1, linewidth=2)
ax.plot(np.arange(1, T + 1), mu_zero, color='b', alpha=1, linewidth=3)
plt.show()
```

Appendix 3. julia 설치 및 R 연동

* 출처 : 윤봉규, “마코프이론 및 응용” 강의자료(국방대학교, 2020.2.16.)

1. Julia 설치

가. Julia 설치 전 여러 언어를 동시에 쓸 수 있도록 Jupyter Notebook을 먼저 설치한다. Jupyter Notebook은 R, Python, Julia 언어를 동시에 사용하여 각 언어의 고유 특징과 장점을 동시에 사용할 수 있다. Jupyter Notebook은 https://www.anaconda.com/distribution/에서 설치파일을 다운받아 관리자모드로 실행하여 설치한다..

나. Julia 홈페이지 (https://julialang.org/downloads/)에서 운영체제에 맞는 Julia 설치파일을 다운받아서 실행한다.

다. Julia 실행 후 “IJulia” 패키지 설치하여 Julia와 Jupyter Notebook을 연결한다.

```
julia> Pkg.add("IJulia")
```

라. Jupyter Notebook 실행 후 Julia를 옵션으로 새 Notbook을 Julia로 실행한다.

2. Julia 에서 R 연동

가. RCall 패키지 설치

```
julia> Pkg.add("RCall")
```

나. 홈디렉토리 설정

```
ENV["R_HOME"]="C:\\Program Files\\R\\R-3.6.2"
```

다. 실행파일 경로 설정

```
ENV["PATH"]="C:\\Program Files|\R\R-3.6.2\\bin"
```

라. Julia에서 R연동 패키지 실행

```
using RCall
```

3. Julia에서 R 실행

```
R"""
rnorm(10)      # R"""과 """ 사이에 R에서 실행하고자 하는 코드 입력
"""
```

Appendix 4. 단계형 분포 julia 코드

* Jupyter Notebook을 활용하여, Julia언어 코드를 기준으로 작성

1. Julia 디렉토리 설정

```
homedir()
cd(raw"C:\ph")
```

2. 관련 패키지 설치 및 실행

```
Pkg.add("Distributions")
Pkg.add("DataFrames")
Pkg.add("EMpht")
Pkg.add("CSV")
Pkg.add("RCall")
Pkg.add("Plots")
Pkg.add("StatsBase")
Pkg.add("VMLS")
Pkg.add("Printf")
Pkg.add("SparseArrays")
Pkg.add("LinearAlgebra")
Pkg.add("QuadGK")

using Distributions, DataFrames, EMpht, CSV, RCall, Plots, StatsBase,VMLS, Printf, SparseArrays, LinearAlgebra, QuadGK
```

3. Moment Matching 방법

```
R"""
setwd("c:/ph")                    # julia와 동일한 디렉토리 설정

install.packages("prcma")         # R패키지 설치
install.packages("tictoc")
install.packages("mapfit")

library(pracma)                   # R패키지 실행
library(tictoc)
library(mapfit)

dat = read.csv("A1.csv")          # CSV파일 불러오기
temp <- dat
x_range=seq(1,200,0.2)
temp=temp[temp>0]

# moment matching method
x1 <- mean(temp)
x2 <- sum(temp^2)/length(temp)
x3 <- sum(temp^3)/length(temp)
(result1 <- phfit.3mom(x1,x2,x3))
ph.moment(3,result1)
c(x1,x2,x3)

pi_c = result1@alpha              #초기확률
T_c = result1@Q                   #일시상태 전이행렬
```

```
a_c = result1@xi                #흡수율
pdff <- function(x){            #단계형 분포 확률밀도함수(PDF)
p_ci%*%expm(T_c*x)%*%a_c
}
"""
data = @rget temp               #R에서 julia로 데이터 불러오기
```

4. 연속형 입력 데이터 EM 알고리즘

```
data = convert(Array{Float64},data)      # data type을 Float로 변환
sample = EMpht.Sample(obs=data)      # 샘플데이터 생성
ph = empht(sample, p=30, ph_structure="CanonicalForm1")
                                         # p : 단계수
ph.π                                     #초기확률
ph.T                                     #일시상태 전이행렬
ph.t                                     #흡수율

function f_c(x)                          # 확률밀도함수(PDF)
    ph.π`*exp(ph.T*x)*ph.t
end
```

5. 이산형 입력 데이터 EM 알고리즘

```
# 데이터를 이산형으로 변환시키기 위한 함수
function bin_observations(data, bins)
      hist = StatsBase.fit(Histogram, data, nbins=bins)
      int = hcat(collect(hist.edges[1][1:end-1]),
      collect(hist.edges[1][2:end]))
```

```
    intweight = convert(Array{Float64}, hist.weights)
    return [hist, int, intweight]
end

# 함수 사용 시 변환할 데이터와 bin값 설정
~, int, intweight = bin_observations(data, 10)
s = EMpht.Sample(int=int, intweight=intweight)
ph_D = empht(s, p=PH, ph_structure="CanonicalForm1")

ph_D.π                                  # 초기확률
ph_D.T                                  # 일시상태 전이행렬
ph_D.t                                  # 흡수율

function f_d(x)                         # 확률밀도함수(PDF)
    ph_D.π'*exp(ph_D.T*x)*ph_D.t
end
```