한국지능시스템학회 논문지 2011, Vol. 21, No. 2, pp. 171-177
DOI : 10.5391/JKIIS.2011.21.2.171

# 기계학습 알고리즘의 컴퓨팅시간 단축을 위한 새로운 통계적 샘플링 기법

# A New Statistical Sampling Method for Reducing Computing time of Machine Learning Algorithms

전성해
Sunghae Jun

청주대학교 바이오정보통계학과

## 요 약

기계학습에서 모형의 정확도와 컴퓨팅시간은 중요하게 다루어지는 부분이다. 일반적으로 모형을 구축하는 데 사용되는 컴퓨팅시간은 분석에 사용되는 데이터의 크기에 비례하여 커진다. 따라서 컴퓨팅시간 단축을 위하여 분석에 사용되는 데이터의 크기를 줄이는 샘플링전략이 필요하다. 하지만 학습데이터의 크기가 작게 되면 구축된 모형의 정확도도 함께 떨어지게 된다. 본 논문에서는 이와 같은 문제를 해결하기 위하여 전체데이터를 분석하지 않아도 전체를 분석할 때와 비슷한 모형성능을 유지할 수 있는 새로운 통계적 샘플링방법을 제안한다. 주어진 데이터의 구조에 따라 최선의 통계적 샘플링기법을 선택할 수 있는 기준을 제시한다. 군집, 층화, 계통추출에 의한 통계적 샘플링기법을 사용하여 정확도를 최대한 유지하면서 컴퓨팅시간을 단축할 수 있는 방법을 보인다. 제안방법의 성능을 평가하기 위하여 객관적인 기계학습 데이터를 이용하여 전체데이터와 샘플데이터 간의 정확도와 컴퓨팅시간을 비교하였다.

## Abstract

Accuracy and computing time are considerable issues in machine learning. In general, the computing time for data analysis is increased in proportion to the size of given data. So, we need a sampling approach to reduce the size of training data. But, the accuracy of constructed model is decreased by going down the data size simultaneously. To solve this problem, we propose a new statistical sampling method having similar performance to the total data. We suggest a rule to select optimal sampling techniques according to given data structure. This paper shows a sampling method for reducing computing time with keeping the most of accuracy using cluster sampling, stratified sampling, and systematic sampling. We verify improved performance of proposed method by accuracy and computing time between sample data and total data using objective machine learning data sets.

**Key Words** : 기계학습 알고리즘, 통계적 샘플링, 컴퓨팅 시간, 군집 샘플링 , 층화 샘플링, 계통 샘플링

## 1. 서 론

기계학습(machine learning)은 관측된 과거의 데이터로부터 학습을 통하여 예측모형을 구축하고, 이를 바탕으로 앞으로 발생되는 여러 문제들에 대하여 최적의 의사결정을 이끌어 내는 방법론이다[1],[2]. 사전에 데이터의 정규성(normality assumption) 가정이 필요한 통계적 분석기법에 비해 대부분의 기계학습 알고리즘은 데이터에 대한 사전가정이 필요하지 않은 유연성 때문에 최근에 전통적인 통계학 분야에서도 사용되고 있다[3],[4],[5],[6]. 기계학습 알고리즘을 이용한 예측모형의 구축은 대용량 데이터베이스에 저장된 데이터의 분석을 통해 이루어진다. 일반적으로 기계학습 알고리즘의 컴퓨팅시간은 분석에 사용되는 데이터의 크기에 비례하여 증가한다. 데이터베이스기술의 개발에 따라 저장된 데이터의 크기는 기하급수적으로 증가하지만 정확도(accuracy)와 같은 예측모형의 성능을 유지하면서 컴퓨팅시간을 단축할 수 있는 방법의 개발은 활발히 이루어지기 않고 있다. 물론 중앙처리장치(CPU)와 같은 컴퓨터 하드웨어의 발전이 이루어지고 있지만 저장되는 데이터용량을 따라잡기는 어렵다. 본 논문에서는 기계학습 알고리즘의 컴퓨팅시간 단축문제의 해결을 위하여 전체데이터의 일부분을 추출하여 분석하는 통계적 샘플링(statistical sampling) 방법을 제안한다. 단순임의추출(simple random sampling)과 같이 기존의 통계적 샘플링 기법을 그대로 적용하여 기계학습의 컴퓨팅시간을 줄이는 연구는 그동안 꾸준히 이루어져 왔다[7],[8],[9],[10],[11]. 전체데이터를 서로 겹치지 않는 층(stratum)들로 나눈 후에 각 층별로 단순임의추출을 수행하는 통계적 샘플링기법을 적용하여 기계학습 컴퓨팅시간의 단축을 얻은 연구결과들도 발표되었다[12],[13],[14],[15]. 이와 같은 기존의 연구결과는 데이터의 형태에 따라 적합한 샘플링 기법을 주관적으로 결정하여 사용하였다. 객관적인 선정기준이 마련되지 못했기 때문에 분석결과의 성능은 분석가의 경험과 지식에 의존하게 된다. 따라서 본 논문에서

접수일자 : 2011년 2월 8일
완료일자 : 2011년 4월 3일
이 논문은 2009학년도 청주대학교가 지원하는 해외파견으로 연구되었음.

는 기존의 주관적 샘플링 전략들과는 다른 객관적인 샘플링 기법을 적용하여 기존의 연구결과에 비해 좀 더 객관적인 방법을 사용하려고 노력하였다. 따라서 본 연구는 기계학습 알고리즘의 컴퓨팅시간 단축을 위하여 객관적인 통계적 샘플링방법을 제안하였다. 제안된 내용의 성능평가를 위하여 UCI Machine Learning Repository로부터의 객관적인 데이터를 이용하여 전체데이터와 샘플데이터 간의 정확도와 컴퓨팅시간을 비교하였다[16].

## 2. 통계적 샘플링

샘플링은 분석대상이 되는 전체 모집단(population) 중 일부를 선택하고 이 부분집단(sample)만을 대상으로 분석을 수행하여 전체 모집단의 특성을 추정(estimation)하는 통계기법이다[17]. 샘플링은 크게 비확률추출(non-probability sampling)과 확률추출(probability sampling)의 2가지 방법으로 나뉜다. 비확률추출은 분석가의 주관적 기준에 의해 샘플을 선정하는 방법이다. 즉, 분석가의 지식과 경험을 통하여 모집단을 가장 잘 나타낸다고 판단되는 개체들을 주관적으로 선정하는 샘플링이다. 반면에 확률추출은 모집단에 속한 모든 개체들에 대하여 사전에 해당되는 추출확률이 결정되는 샘플링 기법이다. 즉, 사전에 각 개체의 추출확률이 결정된 확률추출을 사용하면 선택 가능한 모든 표본의 경우를 알 수 있게 되고 샘플을 통해 계산된 추정량의 통계적 유의성을 확률적으로 계산할 수 있게 된다. 단순임의추출, 층화추출(stratified sampling), 계통추출(systematic sampling), 그리고 군집추출(cluster sampling)은 대표적인 확률추출의 샘플링 기법들이다. 본 논문에서는 확률분포에 의한 추정량의 성능을 평가할 수 있다고 이론적으로 증명되어 있는 객관적인 확률추출을 고려한다. 즉, 확률추출에서는 개개의 샘플에 대응되는 추정값을 계산할 수 있기 때문에 추정값들의 확률분포(probability distribution)를 사용할 수 있고, 이를 통하여 추정량의 표준오차(standard error)를 계산할 수도 있다. 또한 샘플로부터 구한 추정값과 실제 모집단의 모수(parameter)와의 차이인 다음의 샘플링오차(sampling error)를 구할 수 있다[17].

$$Sampling\ Error = \theta - \hat{\theta} \tag{1}$$

여기서 $\theta$는 모수, 그리고 $\hat{\theta}$은 추정값을 나타낸다. 단순임의추출은 크기 N인 모집단에서 크기 n(<<N)인 샘플을 추출할 때 모집단에 속한 모든 개체들의 추출확률을 동일하게 하는 샘플링기법이다. 층화추출은 모집단을 서로 겹치지 않는 층들로 나눈 후에 각 층에 대하여 단순임의추출을 적용하여 샘플을 얻는다. 현재 기계학습을 위한 샘플링기법 적용에 대한 연구결과에서는 단순임의추출과 층화추출이 사용되었다[7],[8],[15],[18],[19]. 층화추출은 모집단을 효과적으로 층화할 수 있을 경우에만 의미가 있다. 만약 제대로 된 층화가 이루어지지 않은 모집단으로부터의 층화추출은 샘플로부터 계산된 추정량의 정도(precision)가 떨어지기 때문에 이 경우에는 단순임의추출보다도 샘플링의 효과가 떨어질 수 있다. 효과적인 층화는 각 층이 내부적으로 동질적(homogeneous)이어야 한다. 층화추출에서는 모집단 전체에 대한 추정뿐만 아니라 각 층별 추정도 가능한 장점이 있다. 층화의 기준이 되는 변수를 층화변수(stratification variable)라고 한다. 적절한 층화변수의 선택은 층화추출의 중요한 요소가 된다. 모집단으로부터 처음 k개의 개체들 중에서 임의로 한 개를 추출하고 이후 k번째 간격마다 한 개의 개체를 연속적으로 추출하는 샘플링기법이 계통추출이다. 계통추출은 샘플링절차가 간편하고 일반적으로 모집단 전체를 잘 반영하는 장점이 있다. 하지만 모집단의 개체들이 어떠한 순서를 가지고 주기적으로 나열되어 있을 경우에는 샘플이 극단적으로 치우칠 수 있는 단점이 있다. 군집추출은 한 개 이상의 개체들로 이루어진 군집이 샘플링의 추출단위가 된다. 앞의 3개의 샘플링기법들의 추출단위는 개체인데 비하여 군집추출은 군집이 추출단위인 점이 다르다. 군집추출에서 추출단위인 군집을 추출할 때에는 단순임의추출을 사용한다. 군집의 크기에 비례하여 해당 군집의 추출확률을 크게 하는 확률비례추출도 있다. 주어진 데이터에 대하여 군집추출을 사용하기 위해서는 해당 데이터에 블록구조가 존재해야 한다.

## 3. 새로운 통계적 샘플링기법을 이용한 기계학습 컴퓨팅 시간단축

최근 통계학 분야에서도 기계학습 알고리즘을 사용하여 보다 우수한 예측모형을 구축하려는 시도가 이루어지고 있다[20]. 하지만 학습시간이 요구되는 기계학습 기법들은 일괄처리(batch) 방식인 통계적 분석기법에 비해 더 많은 컴퓨팅시간을 요구한다[1],[21]. 현재까지 기계학습 알고리즘의 컴퓨팅시간 단축을 위한 여러 시도들이 있고, 본 연구에서는 모형구축을 위하여 사용되는 데이터의 크기를 줄이는 접근을 시도한다. 본 연구에서 고려하는 시도는 통계적 샘플링이다. 기존의 통계적 샘플링 기법들은 주로 여론조사, 마케팅을 위한 시장조사 등 사회과학의 설문조사(survey)에 맞추어 있기 때문에 기계학습 데이터에 그대로 적용하는 데는 어려움이 있다. 본 논문에서는 기계학습 데이터분석에 알맞은 새로운 통계적 샘플링방법을 제안하고 이를 이용한 기계학습 컴퓨팅시간 단축을 위한 방안에 대하여 연구한다. 단순임의추출을 비롯한 군집추출, 층화추출, 계통추출 등의 통계적 샘플링기법들을 이용하여 주어진 데이터에 알맞은 샘플링방법을 개발하여 컴퓨팅시간의 단축을 시도한다. 본 논문에서는 수집된 전체데이터를 모집단으로 설정하고 이것을 가장 잘 대표할 수 있는 샘플을 추출할 수 있는 객관적인 방법을 제안한다. 다음 그림은 본 연구에 대한 전체적인 개념을 간략하게 나타내고 있다.

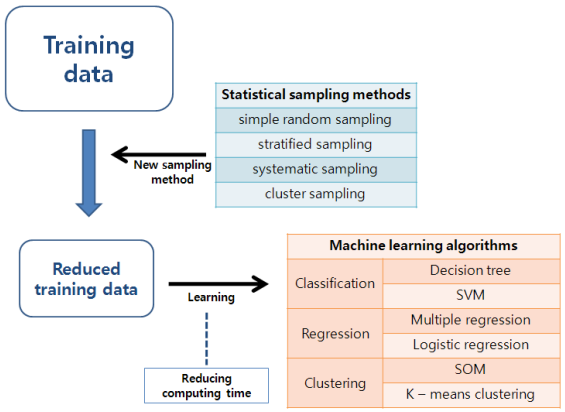


그림 1. 기계학습 계산시간 단축을 위한 통계적 샘플링 절차
Fig. 1. Statistical sampling process for reducing computing time of machine learning

위 그림에서 모집단인 학습데이터는 단순임의추출, 층화추출, 계통추출, 그리고 군집추출의 4가지 통계적 샘플링 기법들을 변형한 새로운 통계적 샘플링방법이 적용된다. 본 논문에서는 의사결정나무모형(decision tree)과 SVM(support vector machine)과 같은 분류모형, 다중선형 회귀모형(multiple linear regression)과 로지스틱 회귀모형(logistic regression)으로 대표되는 회귀모형, 그리고 자기조직화지도(self organizing map, SOM)와 K-평균(means) 군집화와 같은 군집화 모형 등 대부분의 기계학습 알고리즘에 적용할 수 있는 객관적인 샘플링방법을 제안한다. 본 연구에서 기계학습 알고리즘은 샘플로 이루어진 축소된 학습데이터를 이용하여 컴퓨팅시간이 감소된 최적의 모형을 구축하게 된다. 제안된 샘플링방법에 의한 샘플데이터를 학습한 모형은 전체데이터를 사용할 때에 근접한 정확도를 유지하면서 컴퓨팅시간의 단축을 이룰 수 있게 한다. 다음 그림은 본 논문에서 제안하는 통계적 샘플링 선택방법이다.

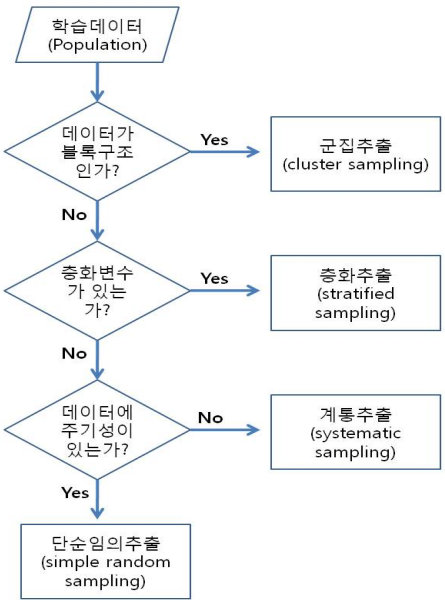


그림 2. 새로운 통계적 샘플링 선택방법
Fig. 2. New selecting method of statistical sampling methods

우선 모집단으로 고려된 학습데이터가 블록(block)구조를 가지고 있는지 확인한다. 일단 전체데이터 안에 블록을 이루는 군집이 존재하면 단순임의추출, 계통추출, 층화추출에 비해 군집추출이 단위비용 당 더 많은 정보를 제공할 수 있기 때문에 우선적으로 군집추출이 고려된다[22],[23]. 데이터가 이와 같은 군집구조를 이루지 않고 있으면 다음으로 고려하게 되는 통계적 샘플링기법은 층화추출이다. 이 추출방법을 사용하기 위해서는 전체데이터를 서로 겹치지 않도록 구분해주는 층화변수가 주어진 데이터에 있어야 한다. 보통 클래스를 나타내는 범주형 변수들 중에서 층화변수를 결정할 수 있다. 이와 함께 각 층을 어떻게 정의할 것인지도 고려의 대상이 된다. 층화추출과 군집추출 간의 근본적인 차이점은 층화추출에서의 층의 특성과 군집추출에서의 군집의 특성이 서로 다른 것이다. 즉, 층화추출에서는 같은 층 내의 개체들은 상대적으로 동질적이어야 하고 층들 간에는 서로 이질적이어야 한다. 군집추출에서는 이와 반대로 같은 군집 내의 개체들은 가능한 이질적이고, 군집들 간은 상대적으로 덜 이질적이어야 효과적인 샘플링 결과를 기대할 수 있다[23]. 따라서 층화추출과 군집추출의 차이를 잘 이해하고 있어야 본 논문에서 제안하는 객관적인 샘플링방법을 실제 기계학습 분석데이터에 효과적으로 적용할 수 있게 된다. 또한 전체데이터를 L개의 층으로 나누는 층화작업을 한 후에 각각의 층을 대상으로 한 군집추출을 수행할 수도 있다. 이와 같은 방법은 군집추출과 층화추출을 결합한 샘플링기법이다. 군집추출과 층화추출을 적용하기 어려운 데이터를 대상으로 본 연구에서 선택할 수 있는 통계적 샘플링기법은 계통추출이다. 계통추출은 단순임의추출에 비해 단순하고 편리하여 더 먼저 고려될 수 있는 샘플링기법이다. 단위비용 당 얻을 수 있는 정보의 양도 단순임의추출보다 더 많다[22]. 군집추출, 층화추출, 그리고 계통추출을 적용하기 어려운 모든 데이터에서는 단순임의추출을 적용한다. 단순임의추출에서 전체데이터에 속한 모든 개체들은 같은 추출가능성을 갖는다. 단순임의추출은 일반적으로 기계학습에서 우선적으로 고려되고 사용되는 대표적인 샘플링기법이다. 지금까지 기계학습 알고리즘에서 고려되어진 붓스트랩(bootstrap), 깁스 샘플러(Gibbs sampler) 등 대부분의 모의실험(simulation) 도구들에서도 단순임의추출이 기본적으로 사용되었다[24]. 결론적으로 본 연구에서 기계학습 컴퓨팅시간 단축을 위하여 제안하는 통계적 샘플링방법은 주어진 데이터의 구조를 파악하여 우선적으로 군집추출을 사용하고 이 추출방법의 적용이 어려울 경우에 다음으로 고려될 수 있는 샘플링방법은 층화추출, 계통추출의 순서로 이루어진다. 층화추출과 계통추출도 이 추출방법들이 적용될 수 있는 조건들을 잘 살펴야 한다. 물론 군집, 층화, 계통추출의 샘플링방법의 적용이 모두 어려운 데이터에 대해서는 단순임의추출을 사용해야 할 것이다.

## 4. 실험 및 결과

기계학습 컴퓨팅시간 단축을 위하여 본 논문에서 제안하는 통계적 샘플링방법의 성능평가 실험을 위하여 UCI machine learning repository로부터 객관적인 기계학습 데이터를 이용하였다[16]. 실험에서 사용될 기계학습 알고리즘으로 본 연구에서는 대표적인 기계학습 모형인 신경망(neural networks)에 본 연구의 제안방법을 적용하였다[2]. 본 실험에서 사용된 통계적 샘플링방법과 기계학습 모형구축 및 컴퓨팅시간 계산을 위한 분석도구로는 통계계산(statistical computing) 분야에서 주로 사용되는 R 언어를 이용하였다[25]. 군집추출, 층화추출, 계통추출, 그리고 단순임의추출을 위하여 R에서 제공하는 'sampling' 패키지를 이용하였고[26], 신경망모형을 위하여 역시 R의 'nnet' 패키지를 이용하였다[27]. 첫 번째 데이터는 Census Income Data Set이다[16]. 14개의 입력변수와 1개의 출력변수를 포함하고 있다. 나이(age), 학력(education-num) 등에 따라 2개의 수준(<=50K, >50K)을 갖는 연봉을 예측하고 분류하는 작업이다. 이 데이터는 'native-country' 변수에 의해 블록화 되어 있다. 전체 32,561개의 각 개체는 이 블록변수에 의해 전체 블록레이블 중 하나에 소속되어 있다. 본 실험에서는 41개의 블록 레이블들 중에서 10개의 레이블을 군집추출하였다. 아래의 표는 Census Income 데이터의 블록구조에 대한 전체와 추출된 레이블의 결과를 나타낸다.

표 1. 전체 블록레이블과 추출된 블록레이블
Table 1. All block labels and sampled labels

| All labels in block (all # = 41) | Sampled labels in block (selected # = 10) |
|---|---|
| United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands | El-Salvador, France, Italy, Mexico, Nicaragua, Peru, Scotland, Taiwan, Trinadad&Tobago, Vietnam |

41개의 블록은 각 출신국가를 나타낸다. 출신국가 내에는 나이, 교육연수 등 다양한 특성을 가진 이질적인 개체들이 존재하기 때문에 군집추출의 특성을 만족한다. 이와 같은 추출결과를 이용하여 최종적으로 모형구축에 사용되는 데이터의 개수는 다음 표와 같다.

표 2. 군집추출 결과
Table 2. Result of cluster sampling

| Data | Training | | Test |
|---|---|---|---|
| | Total | Sample | |
| # of instances | 32,561 | 1,113 | 16,281 |
| # of clusters in sample=10 | | | |

위의 결과에서 추출된 샘플데이터는 모두 1,113개이다. 이는 전체 데이터의 3.4%에 해당된다. 이 데이터는 모두 'native-country' 변수의 레이블이 El-Salvador, France, Italy, Mexico, Nicaragua, Peru, Scotland, Taiwan, Trinadad&Tobago, Vietnam에 해당되는 개체들이다. 전체 데이터와 군집추출된 샘플데이터에 대하여 신경망을 이용하여 분류모형을 구축하였다. 입력변수로는 5개의 연속형변수들(age, fnlwgt, education-num, capital-gain, capital-loss, hours-per-week)을 이용하였고, 1개의 출력변수는 2개(<=50K, >50K)의 클래스를 가지고 있다. 본 실험에서 사용된 신경망모형은 다층퍼셉트론(multi-layer perceptron)이고 하나의 은닉층(hidden layer)을 가지며 은닉층의 노드(node)수는 10으로 하였다. 초기 가중치(initial weight)는 1로 하였고 최대 반복수(maximum number of iterations)는 100으로 하였다. 다음 표는 전체데이터를 사용한 신경망모형과 군집추출에 의한 샘플데이터를 사용한 신경망모형의 정확도(accuracy)와 계산시간(computing time)에 대한 비교결과를 나타내고 있다.

표 3. 전체데이터와 군집추출 샘플데이터의 성능평가
Table 3. Performance comparison between total data and cluster sampling data

| | Total | Cluster sampling |
|---|---|---|
| Accuracy (%) | 76.58 | 76.38 |
| Computing time(second) | 5.84 | 0.5 |

위의 결과를 통하여 군집추출을 적용하여 전체데이터의 3.4% 만으로 이루어진 샘플데이터와의 정확도 차이는 각각 76.58%과 76.38%로 거의 차이를 보이지 않고 있음을 알 수 있다. 하지만 계산시간은 각각 5.84초와 0.5초로 12배에 가까운 차이를 보이고 있다. 특히 샘플데이터의 계산시간 0.5초에는 군집추출 작업에서 사용된 0.29초를 합친 결과이다. 즉 신경망모형 분석시간 0.21초와 군집추출 작업시간 0.29초가 합쳐진 시간이다. 따라서 주어진 데이터가 블록구조를 이루고 있을 경우에 우선적으로 군집추출에 의한 통계적 샘플링을 수행하고, 이렇게 얻어진 샘플데이터를 이용한 모형구축을 할 경우에 전체데이터를 이용했을 경우와 비교하여 정확도인 모형의 성능은 큰 차이를 보이지 않으면서 계산시간은 단축할 수 있음을 확인할 수 있었다. 층화추출에 의한 기계학습 알고리즘의 컴퓨팅시간 단축은 기존의 연구들에서 이미 확인되었다[18],[19]. 마지막으로 계통추출에 의한 신경망모형의 계산시간 단축을 위한 실험을 위하여 UCI machine learning repository로부터의 Magic Gamma Telescope Data Set을 이용하였다[16]. 이 데이터는 10개의 연속형 입력변수들과 2개의 범주를 갖는 1개의 출력변수로 이루어져 있다. 19,020 개로 이루어진 전체데이터 중에서 2/3는 학습데이터로 그리고 1/3은 테스트데이터로 사용하였다[2]. 실험을 위하여 20%, 30%, 40%, 그리고 50%의 계통추출 샘플을 이용하였다. 다음 표는 실험에 사용된 학습, 샘플, 그리고 테스트 데이터의 크기를 나타낸다.

표 4. 학습, 샘플, 테스트 데이터
Table 4. Training, sample, and test data sets

| Data | | # of instances |
|---|---|---|
| Total | | 12,743 |
| Sample | 20% | 2,549 |
| | 30% | 3,823 |
| | 40% | 5,097 |
| | 50% | 6,372 |
| Test | | 6,277 |

우선 전체데이터를 이용한 신경망모형의 정확도와 컴퓨팅시간은 다음과 같다.

표 5. 전체테스트의 분석결과
Table 5. Result of total data analysis

| Accuracy (%) | Computing time(second) |
|---|---|
| 82.27 | 18.16 |

위 표는 12,743개의 데이터를 이용하여 앞의 Census Income Data Set를 이용한 실험과 동일한 신경망모형을 구축하였다. 구축된 모형을 적용하여 6,277개의 테스트데이

터에 대한 정확도를 조사한 결과를 나타내고 있다. 계통추출에 의한 기계학습모형을 위한 컴퓨팅시간 단축을 위하여 본 실험에서는 샘플링시간(sampling time), 신경망모형구축시간(modeling time), 그리고 샘플데이터로부터 구축된 모형을 적용하여 테스트데이터로부터 구한 정확도를 비교하였다. 샘플데이터로부터의 샘플링시간과 모형구축시간을 더한 값과 전체데이터의 컴퓨팅시간의 차이를 비교하였다. 다음 4개의 표들은 20%, 30%, 40%, 그리고 50%의 계통추출에 의한 결과들이다.

표 6. 샘플테스트의 분석결과 (20%)
Table 6. Result of sample data analysis (20%)

| Repeat | S. time | M. time | T. time | Accuracy |
|---|---|---|---|---|
| 1 | 0.07 | 1.69 | 1.76 | 0.7873 |
| 2 | 0.11 | 1.89 | 2.00 | 0.7360 |
| 3 | 0.10 | 1.84 | 1.94 | 0.6959 |
| 4 | 0.10 | 1.78 | 1.88 | 0.7518 |
| 5 | 0.07 | 1.85 | 1.92 | 0.7798 |
| 6 | 0.11 | 1.87 | 1.98 | 0.7634 |
| 7 | 0.09 | 2.08 | 2.17 | 0.7144 |
| 8 | 0.06 | 2.00 | 2.06 | 0.6970 |
| 9 | 0.11 | 1.98 | 2.09 | 0.7394 |
| 10 | 0.13 | 2.03 | 2.16 | 0.7661 |

표 7. 샘플테스트의 분석결과 (30%)
Table 7. Result of sample data analysis (30%)

| Repeat | S. time | M. time | T. time | Accuracy |
|---|---|---|---|---|
| 1 | 0.10 | 5.49 | 5.59 | 0.7166 |
| 2 | 0.11 | 5.50 | 5.61 | 0.7948 |
| 3 | 0.08 | 5.23 | 5.31 | 0.8126 |
| 4 | 0.03 | 5.77 | 5.80 | 0.8031 |
| 5 | 0.11 | 5.82 | 5.93 | 0.8071 |
| 6 | 0.11 | 5.80 | 5.91 | 0.7875 |
| 7 | 0.06 | 5.73 | 5.79 | 0.7999 |
| 8 | 0.07 | 4.32 | 4.39 | 0.7801 |
| 9 | 0.11 | 6.32 | 6.43 | 0.7746 |
| 10 | 0.08 | 5.74 | 5.82 | 0.7789 |

표 8. 샘플테스트의 분석결과 (40%)
Table 8. Result of sample data analysis (40%)

| Repeat | S. time | M. time | T. time | Accuracy |
|---|---|---|---|---|
| 1 | 0.11 | 5.96 | 6.07 | 0.7983 |
| 2 | 0.06 | 7.37 | 7.43 | 0.7682 |
| 3 | 0.09 | 7.00 | 7.09 | 0.7582 |
| 4 | 0.09 | 7.78 | 7.87 | 0.7937 |
| 5 | 0.11 | 7.73 | 7.84 | 0.7880 |
| 6 | 0.09 | 6.46 | 6.55 | 0.7822 |
| 7 | 0.06 | 7.64 | 7.70 | 0.7242 |
| 8 | 0.11 | 7.17 | 7.28 | 0.7398 |
| 9 | 0.06 | 7.95 | 8.01 | 0.7822 |
| 10 | 0.11 | 7.29 | 7.40 | 0.7555 |

표 9. 샘플테스트의 분석결과 (50%)
Table 9. Result of sample data analysis (50%)

| Repeat | S. time | M. time | T. time | Accuracy |
|---|---|---|---|---|
| 1 | 0.08 | 7.57 | 7.65 | 0.8085 |
| 2 | 0.09 | 9.75 | 9.84 | 0.7770 |
| 3 | 0.09 | 9.25 | 9.34 | 0.7856 |
| 4 | 0.08 | 9.80 | 9.88 | 0.7682 |
| 5 | 0.11 | 7.72 | 7.83 | 0.7970 |
| 6 | 0.11 | 8.69 | 8.80 | 0.7754 |
| 7 | 0.08 | 10.58 | 10.66 | 0.7876 |
| 8 | 0.09 | 10.14 | 10.23 | 0.7876 |
| 9 | 0.09 | 10.30 | 10.39 | 0.7945 |
| 10 | 0.05 | 8.94 | 8.99 | 0.7722 |

각 샘플수준(20%-50%)에서 10번의 반복실험을 수행하였다. S. time과 M. time은 샘플링시간과 모형구축시간을 나타낸다. T. time은 S. time과 M. time을 합한 값이다. 전체데이터의 컴퓨팅시간과 비교되는 시간은 T. time이다. 다음 표는 모든 샘플수준에서의 컴퓨팅시간과 정확도에 대한 분포를 나타내고 있다.

표 10. 전체 샘플테스트의 분석결과
Table 10. Result of all sample data sets

| Sample | Ave. Time | Accuracy | | | |
|---|---|---|---|---|---|
| | | Mean | Min | Max | S.D. |
| 20% | 1.996 | 0.7431 | 0.6959 | 0.7873 | 0.0326 |
| 30% | 5.658 | 0.7440 | 0.6787 | 0.7697 | 0.0259 |
| 40% | 7.324 | 0.7690 | 0.7242 | 0.7983 | 0.0243 |
| 50% | 9.361 | 0.7854 | 0.7682 | 0.8085 | 0.0125 |

전체데이터의 컴퓨팅시간이 18.86초인데 비해 샘플데이터의 컴퓨팅시간은 20% 계통추출에서는 평균 컴퓨팅시간이 1.996초로 훨씬 단축되었으며, 50% 계통추출에서도 9.361초로 많이 단축된 결과를 얻었다. 정확도는 계통추출의 샘플크기가 커짐에 따라 전체데이터의 정확도에 근접하고 있음을 알 수 있다. Mean, Min, Max, 그리고 S.D.는 각각 정확도의 평균, 최소, 최대, 그리고 표준편차를 나타낸다. 모든 샘플수준에서 작은 표준편차값이 계산됨으로서 제안방법의 안정성도 확인할 수 있었다. 계통추출의 샘플크기는 허용할 수 있는 정확도 수준과 컴퓨팅시간을 고려하여 분석가에 의해 결정될 수 있다.

선행연구들[18],[19]과 본 논문의 실험결과에 의하면 군집추출에 의한 실험결과에서 가장 좋은 성과를 얻었으며 다음으로 층화추출, 계통추출의 순이었다. 따라서 본 논문에서 제안하는 방법에 의해 데이터의 블록구조를 확인하여 우선적으로 군집추출을 고려하고 이것이 만족되지 않을 경우 층화변수의 사용 가능성 여부에 의한 층화추출을 그리고 마지막 단계로 데이터의 주기성을 파악하여 계통추출을 적용할 수 있을 것이다. 물론 군집, 층화, 계통추출이 모두 어려울 경우에는 기존의 단순임의추출을 사용하면 된다.

## 5. 결 론

본 논문에서는 전체 데이터를 이용한 학습모형과 비슷한

정확도를 유지하면서 동시에 컴퓨팅시간을 줄일 수 있는 축소된 학습데이터를 생성하는 통계적 샘플링방법을 제안하였다. 전체데이터의 구조를 단계별로 확인하면서 가장 적절한 통계적 샘플링기법을 객관적으로 결정할 수 있게 하였다. 처음 주어진 데이터에 블록구조가 있으면 군집추출을 수행하고 군집추출이 어려울 경우에는 다음으로 층화변수가 있는지 확인한다. 층화변수가 있으면 층화추출에 의한 통계적 샘플링을 수행한다. 군집추출과 층화추출이 되지 않을 경우에는 데이터의 주기성을 확인하여 주기성이 발견되지 않으면 샘플수준을 결정하면서 계통추출을 수행한다. 군집추출, 층화추출, 그리고 계통추출이 모두 어려운 경우에는 마지막으로 일반적으로 기계학습에서 사용되는 단순임의추출을 적용하면 될 것이다. 그러므로 데이터의 수집과 사전처리 단계부터 본 논문에서 제안하는 데이터의 블록구조, 층화변수, 주기성 등을 고려한다면 다양한 기계학습 작업에서 컴퓨팅시간 단축을 위한 통계적 샘플링방법을 기대할 수 있을 것이다. 좀 더 발전된 샘플링 전략으로 층화, 군집, 계통추출 등 기존의 통계적 샘플링기법을 결합한 하이브리드 샘플링기법에 대한 연구도 고려한다. 이를 통해 작은 샘플데이터만으로도 전체데이터를 분석한 정확도에 매우 근접하는 결과를 기대할 수 있게 된다. 이는 향후 연구과제로 남긴다.

## 참 고 문 헌


[1] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer, 2001.

[2] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[3] A. Ben-Hur, D. Horn, H. T. Siegelmann, V. N. Vapnik, “Support Vector Clustering,” *Journal of Machine Learning Research,* vol. 2, pp. 125-137, 2001.

[4] S. R. Gunn, “Support Vector Machines for Classification and Regression,” *Technical Report*, University of Southampton, 1998.

[5] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[6] V. N. Vapnik, “An Overview of Statistical Learning Theory,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988-999, 1999.

[7] Z.-J. Chen, B. Liu, X.-P. He, “A SVC Iterative Learning Algorithm Based on Sample Selection for Large Samples,” *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3308-3313, 2007.

[8] M.-H. Ha, L.-F. Zheng, J.-Q. Chen, “The Key Theorem of Learning Theory Based on Random Sets Samples,” *Proceedings of International Conference on Machine Learning and Cybernetics,* vol. 5, pp. 2826-2831, 2007.

[9] Y. S. Jia, C. Y. Jia, H. W. Qi, “A New Nu-Support Vector Machine for Training Sets with Duplicate Samples,” *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, pp. 4370-4373, 2005.

[10] W. Ng, M. Dash, “An Evaluation of Progressive Sampling for Imbalanced Data Sets,” *Proceedings of Sixth IEEE International Conference on Data Mining*, pp. 657-661, 2006.

[11] K.-H. Yang, G.-L. Shan L.-L. Zhao, “Correlation Coefficient Method for Support Vector Machine Input Samples,” *Proceedings of International Conference on Machine Learning and Cybernetics*, pp. 2856-2861, 2006.

[12] C. S. Ding, Q. Wu, C. T. Hsieh, M. Pedram, “Stratified Random Sampling for Power Estimation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 6, pp. 465-471, 1998.

[13] M. Keramat, R. Kielbasa, “A study of stratified sampling in variance reduction techniques for parametric yield estimation,” *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 1652-1655, 1997.

[14] P. A. D. I. Santos, Jr., R. J. Burke, J. M. Tien, “Prograssive Random Sampling With Stratification,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Applications and Reviews*, vol. 37, no. 6, pp. 1223-1230, 2007.

[15] M. Xing, M. Jaeger, H. Baogang, “An Effective Stratified Sampling Scheme for Environment Maps with Median Cut Method,” *Proceedings of International Conference on Computer Graphics, Imaging and Visualisation*, pp. 384-389, 2006.

[16] The UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml/

[17] S. K. Thompson, *Sampling*, 2nd ed., John Wiley & Sons, 2002.

[18] S. Jun, “Support Vector Machine based on Stratified Sampling,” *International Journal of Fuzzy Logic and Intelligent System*, vol. 9, no. 2, pp. 141-146, 2009.

[19] S. Jun, “Improvement of SOM using Stratifiation,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 9, no. 1, pp. 36-41, 2009.

[20] S. Jun, “Web Usage Mining Using Evolutionary Support Vector Machine,” *Lecture Note in Artificial Intelligence*, vol. 3809, pp. 1015-1020, Springer-Verlag, 2005.

[21] J. Wang, X. Wu, C. Zhang, “Support vector machines based on K-means clustering for real-time business intelligent systems,” *International Journal Business Intelligence and Data Mining*, vol. 1, no. 1, pp. 54-64, 2005.

[22] 김영원, 류제복, 박진우, 홍기학 역, *표본조사의 이해와 활용*, 교우사, 2006.

[23] R. L. Scheaffer, W. Mendenhall III, R. L. Ott, *Elementary Survey Sampling* 6th edition, Duxbury, 2006.

[24] 손건태, *전산통계개론 - 통계적 모의실험과 추정 알고리즘* 제4판, 자유아카데미, 2005.

[25] R Development Core Team, R: A language and en-

vironment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, http://www.R-project.org, 2010.
[26] Y. Tille, A. Matei, *Survey Sampling-Package 'sampling'*, R-Project CRAN, 2009.
[27] B. Repley, *Feed-forward Neural Networks and Multinomial Log-Linear Models-Package 'nnet'*, R-Project CRAN, 2009.

## 저 자 소 개

**전성해(Sunghae Jun)**
1993년 : 인하대 통계학과(학사)
1996년 : 인하대 통계학과(이학석사)
2001년 : 인하대 통계학과(이학박사)
2007년 : 서강대학교 컴퓨터공학과 (공학박사)
2003년～현재 : 청주대학교 바이오정보통계학과 부교수

관심분야 : 기술경영, 인공지능, 데이터마이닝
Phone : 043-229-8205
Fax : 043-229-8432
E-mail : shjun@cju.ac.kr