# 입력 데이터의 분포를 고려한 가상 샘플 생성

이봉기[0] 임용업 조성준
서 울 대 학 교 산 업 공 학 과
{bklee77, ultpblu0, zoon}@snu.ac.kr

# Virtual Samples Generation Based on the Distribution of Input Data

Bong-Ki Lee[0] Yong-Eop Lim Sungzoon Cho
Dept. of Industrial Engineering, Seoul National University

**요 약**

본 논문에서는 잡음 추가와 네트웍 앙상블을 이용하는 기법으로 최근에 제안된 가상 샘플 생성 방법(VSG:Virtual Sample Generation)을 개선하는 방법을 제안하고, 이를 대표적인 앙상블 학습 알고리즘인 Bagging, Boosting과 비교한다. 기존의 가상 샘플 생성 방법에 기초하여 입력 데이터의 분포를 고려하여 가상 샘플을 생성하는 방법을 제안한다. 이 방법은 입력 분포의 밀도가 높은 곳에서 가상 샘플로 인한 과소 적합을 방지하고 밀도가 낮은 곳에서 가상 샘플로 인한 과도 적합을 방지하기 위한 것이다.

본 논문은 입력 데이터의 밀도를 추정하는 새로운 과정을 정리하고 입력 분포에 따라 적합한 가상 샘플을 생성하는 방법을 고안했다. 그리고 제안하는 방법의 일반화 성능 향상을 보이기 위해 여러 가지의 합성 데이터를 사용하여 실험을 하였고 이를 Bagging, Boosting, VSG의 성능과 비교하였다.

## 1. 서론

주어진 패턴을 학습하여 비선형 함수를 근사하는 보편 근사기(universal function approximator) 로는 MLP(Multilayer Perceptron) 과 RBF(Radial Basis Function) 네트웍 등이 있다. 이 보편 근사기를 이용하여 일반화 성능을 높이는 방법으로 앙상블 학습 알고리즘이 있다. 이 기법으로는 Bagging(Bootstrap Aggregating), Boosting과 최근에 제안된 가상 샘플 생성 방법(VSG:Virtual Sample Generation)이 있다. VSG는 주어진 학습 데이터로부터 인공적으로 가상 샘플을 생성하여 학습에 참여시키는 방법이다. 우선 학습 데이터로 신경망들을 학습시킨 후, 이 신경망들의 앙상블로 가상 샘플의 출력 패턴을 결정한다. 이 VSG에 아동간의 관찰 학습이라는 개념을 도입한 것이 관찰 학습 알고리즘(OLA: Observa tional Learning Algorithm)이다[6]. 하지만 OLA는 다른 앙상블 방법들에 비해 학습 시간이 오래 걸리는 단점이 있다. 반면 VSG는 가상 샘플을 추가하여 한번만 다시 학습하므로 학습 시간에 거의 차이가 없다.

VSG는 주어진 데이터의 입력에 잡음을 주어 가상 샘플의 입력값을 만든 후 미리 학습된 신경망들의 앙상블을 이용해 출력 패턴을 결정한다. 그러나 주어진 학습 데이터의 분포가 불균일한 경우, 가상 샘플이 과도 적합과 과소 적합의 원인으로 작용할 수 있다. 따라서 학습 데이터의 분포를 고려하여 학습에 적합한 가상 샘플을 생성하여야 한다.

본 논문에서는 학습 데이터의 입력 분포의 밀도에 따라 가상 샘플의 생성을 다르게 하는 방법을 제안한다. 그리고 제안하는 방법이 기존의 VSG보다 일반화 성능이 우수함을 실험적으로 보이고자 한다.

2장에서는 관련 연구를 소개하고, 3장은 실험에 적용한 VSG와 입력 분포를 고려하는 VSG를 설명하고, 4장은 실험에 사용한 합성 데이터의 생성 방법과 실험 내용을 설명한다. 5장은 앙상블 학습 알고리즘간에 실험 결과를 비교하고, 6장은 본 논문의 목적과 결론을 요약하고 토의점과 추후 연구과제를 제시한다.

## 2. 관련 연구

일반화 성능을 향상시키는 기법으로는 잡음 추가와 앙상블 학습 알고리즘이 있다. 잡음 추가는 학습 데이터의 입력에 잡음을 주는 것으로 과도 학습을 방지하면서 정규화와 비슷한 역할을 한다. 앙상블 학습 알고리즘은 같은 문제를 여러 네트웍을 이용하여 학습한 후, 이들을 결합하여 최종 답을 내는 것이다. 이 때 여러 네트웍들간에 상관 관계를 줄이기 위해 데이터의 재추출 방법을 이용한 것이 Bagging이다. Bagging은 주어진 데이터를 bootstrap하여 여러 개의 학습 데이터를 만들고 각 네트웍을 서로 다른 학습 데이터로 학습한 후, 그들의 결과를 결합하는 방법이다[5].

위의 기법들과는 달리 학습 패턴 수를 늘리는 것이 가상 샘플 생성 방법(VSG)이다. 가상 샘플의 출력 패턴을 결정하는 방법은 여러 가지가 있는데 모든 네트웍의 출력값에 가중치를 둔 평균을 이용할 수도 있고, 가장 학습을 잘한 네트웍을 이용할 수도 있다. 또 하나의 입력에 대해 각 네트웍의 출력들을 모두 이용하여 여러 개의 가상 샘플을 생성할 수도 있다. VSG의 종류로는 단순 VSG, Bootstrap VSG, 선별 VSG, 검증 VSG가 있다 [5].

### 3. 가상 샘플 생성 방법과 입력 분포 고려 방법

본 논문은 여러 가지의 가상 샘플 생성 방법 중 Bootstrap VSG 를 개선 대상으로 한다. Bo otstrap VSG는 Bagging과 같이 주어진 데이터를 bootstrap하여 여러 개의 학습 데이터를 만든다. 우선 이 데이터들을 각각 다른 네트웍으로 학습하고, 정규 분포를 이용하여 입력에 잡음을 주어서 가상 샘플을 생성한다. 가상 샘플을 학습 데이터에 추가하여 다시 학습한 후, 각 네트웍을 앙상블하여 최종 결과를 낸다[4]. 이 알고리즘은 그림 1에 정리되어 있다.

입력 분포를 고려하는 가상 샘플 생성 방법(DVSG:Distribution-based VSG) 은 Bootstrap VSG 의 알고리즘에서 가상 샘플을 생성하는 2단계만을 바꾼 것이다. 이는 그림 2에 정리되어 있다.

DVSG는 입력 분포의 밀도가 높은 곳에서는 가상 샘플의 입력에 잡음을 작게 주고, 입력 분포의 밀도가 낮은 곳에서는 가상 샘플의 입력에 잡음을 크게 준다. 이는 밀도가 높은 곳에서 가상 샘플로 인한 과소 적합을 방지하고 밀도가 낮은 곳에서 가상 샘플로 인한 과도 적합을 방지하기 위한 것이다. 즉 밀도가 낮은 곳에서는 학습의 smoothness를 더 높이고 밀도가 높은 곳에서는 주어진 패턴을 더 정확히 학습하게 하는 방법이다.

우선 주어진 학습 데이터의 각 입력 분포의 확률 밀도를 계산한다. 밀도를 계산하기 위해 비모수적 밀도 추정법의 하나인 Kernel-based method를 이용하고 정규 함수를 kernel function 으로 사용한다. 이 때 정규 함수의 smoothing parameter는 K-nearest neighbors를 이용하는데 0과 1사이라는 확률 밀도의 조건을 만족시키는 값 중에서 최대값으로 구한다[1]. 이는 비모수적 밀도 추정에서 kernel의 크기가 작을수록 바람직하기 때문이다[2]. 이렇게 구해진 밀도 추정치는 가상 샘플의 입력에 주는 잡음의 크기를 조정하는데 쓰인다. 밀도의 크기에 반비례하게 잡음의 크기를 조정하기 위해 밀도 추정치를 역으로 변환시켜 잡음의 분산(Σ)에 곱한다. 이 때 잡음에는 상한과 하한이 있고 그 크기를 조정할 수 있다. 이는 밀도에 따른 잡음의 크기가 입력 데이터의 차원 수와 변수 간의 상관성에 의해 과도하게 커지거나 작아짐으로 인한 잘못된 학습을 방지하기 위한 것이다.

Let $\{f_i \mid i=1,...,L\}$ be a set of networks in the ensemble.

Let $\{D_i \mid i=1,...,L\}$ be the bootstrapped data sets from the original data set $D$.

1. Train each networks, $f_i(\vec{x},\vec{y})$ where $(\vec{x},\vec{y})\in D_i$.

2. For each network, virtual data sets are generated as follows :

$$V_i = \left\{ \left(\vec{v}_j, \bar{f}_i(\vec{v}_j)\right) \middle| \vec{v}_j \sim N(\vec{x}_{ij}, \Sigma), \vec{x}_{ij} \in D_i, j=1,...,n \right\}$$

where $\bar{f}_i = \frac{1}{L}\sum_{j=1}^{L} f_j$

3. Combine $D_i$ and $V_i$ : $D_i = D_i \cup V_i$

4. Initialize each network, and train them again.

5. Obtain the outputs by simple averaging : $\bar{f} = \frac{1}{L}\sum_{i=1}^{L} f_i$

그림 1. Bootstrap 가상 샘플 생성 방법

2. For each network,

1) Set the size of virtual sample's difference, $S$.

2) By enough experiments, decide the nearest neighbor number, $K$ as the maximum number that satisfies :

$$\frac{1}{L}\sum_{i=1}^{L} \max_j\{\hat{p}(\vec{x}_{ij})\} \le 1, \quad \hat{p}(\vec{x}_{ij}) = \frac{1}{n(2\pi)^{d/2}\sigma^d}\sum_{q=1}^{n}\exp\left\{-\frac{|\vec{x}_{ij}-\vec{x}_{iq}|^2}{2\sigma^2}\right\}$$

and $\sigma = \frac{1}{nK}\sum_{j=1}^{n}\sum_{\forall \vec{x}_{iv}\in Knn}(\vec{x}_{ij}-\vec{x}_{iv})^2$ ( $Knn$: K-nearest neighbors of $\vec{x}_{ij}$ )

3) Virtual data sets are generated as follows :

$$V_i = \left\{ \left(\vec{v}_j, \bar{f}_i(\vec{v}_j)\right) \middle| \vec{v}_j \sim N(\vec{x}_{ij}, M(\vec{x}_{ij})\Sigma), \vec{x}_{ij} \in D_i, j=1,...,n \right\}$$

where $\bar{f}_i = \frac{1}{L}\sum_{j=1}^{L} f_j$, $M(\vec{x}_{ij}) = \begin{cases} 1+S & \text{if } M > 1+S \\ 1-S & \text{if } M < 1-S \\ M & \text{otherwise} \end{cases}$

and $M = 1+S-2S*\hat{p}(\vec{x}_{ij})$

그림 2. 입력 분포를 고려한 가상 샘플 생성 방법 중 2단계

### 4. 실험 데이터

실험은 MLP, Bagging, Boosting, 그리고 VSG와 DVSG의 일반화 성능 비교를 위해 특정 함수에서 인공적으로 데이터를 생성한 합성 데이터를 사용한다.

합성 데이터는 입력과 출력이 모두 1차원이고, -1에서 1사이의 입력 X에 대해 $Y=\sin 3(X+0.8)^2+\varepsilon$의 식으로 출력 Y를 계산한다. 여기서 ε는 출력 패턴의 잡음으로 평균이 0인 정규 분포를 따르는 값이다. 즉 이 합성 데이터는 2차원상에서 목표 함수를 학습하는 회귀 문제이다.

여러 경우에 대해 충분한 실험을 하기 위해 6가지의 합성 데이터를 사용하였다. 우선 출력에 주는 잡음(ε)의 분산을 0.01로 정하여 동일한 간격으로 입력이 균일한 데이터와 입력이 랜덤한 데이터, 그리고 두 극점 근처의 입력의 밀도가 높은 데이터의 3가지 데이터를 각 30개씩 생성하였다. 그리고 ε의 분산을 0.16으로 정하여 입력을 랜덤하게 준 3가지의 데이터를 각 15개씩 생성하였다. 그림 3부터 그림 8은 실험에 사용된 6가지 데이터의 학습 패턴과 목표 함수를 나타낸 것이다.

모든 데이터에 대해 앙상블 네트웍의 수는 30개이고, DVSG에서 생성한 가상 샘플 수도 30개이다. 각 데이터에 대해 은닉층이 1개인 MLP를 10 epoch 만큼 Levenberg-Marquardt 알고리즘으로 학습시켜서 최적의 은닉 노드 수를 정하고, DVSG에서 알고리즘의 조건을 만족시키는 Nearest neighbor 수를 정한다. Bagging에서 최종 출력의 결정은 모든 네트웍의 결과를 단순 평균해서 구하고, Boosting 은 Adaboost.R2 알고리즘을 이용한다[3].

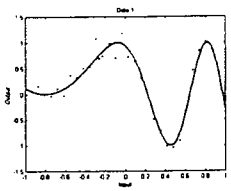

그림 3. Data 1

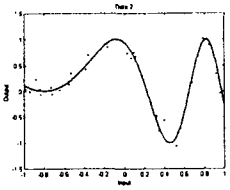

그림 4. Data 2

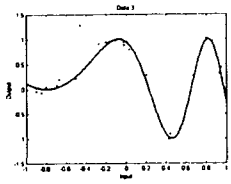

그림 5. Data 3

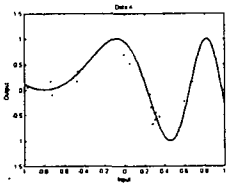

그림 6. Data 4

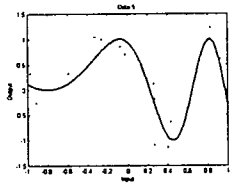

그림 7. Data 5

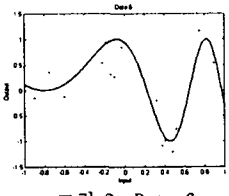

그림 8. Data 6

## 5. 실험 결과

6가지의 합성 데이터에 대해 5가지 기법을 적용하였다. 각 30회의 실험을 하여 평균제곱오차를 평균한 결과와 평균제곱오차에 대한 분산이 표 1에 나와있다. 표 1에서 AMSE는 Averaged Mean Squared Error, VMSE는 Variance of MSE, BAG 은 Bagging, 그리고 BOOST는 Boosting이다.

실험 결과를 보면 Data 2와 Data 5에서 DVSG가 일반화 성능이 가장 좋고, Data 1에서는 Bagging과 DVSG의 성능에 차이가 없고, Data 3에서는 VSG와 DVSG간에 차이가 없다. Data 4 와 Data 6 에서는 VSG 와 Bagging이 좋은 성능을 보였다. MSE의 분산은 MLP와 Boosting이 크고 나머지 기법은 차이가 없이 매우 작으므로 Bagging과 VSG, DVSG 의 성능이 매우 안정적임을 알 수 있다.

실험 결과에서 전체적으로는 Bagging과 VSG, DVSG가 성능이 좋았고, 잡음이 작은 경우에는 DVSG가 성능이 우수한 반면, 잡음이 큰 경우에는 Bagging과 VSG의 성능이 안정적으로 좋았다. 즉 DVSG는 잡음에 다소 민감한 특징이 있는데 이는 밀도가 높은 곳의 잡음이 많은 데이터를 DVSG가 VSG보다 더 정확히 학습하기 때문으로 해석된다.

그림 9는 Data 5에 5가지 기법을 적용한 결과 중 하나이다. Data 5의 결과만을 보여주는 것은 Data 1,2,3의 경우 오차가 작아서 선들이 겹쳐 보이기 때문이다. Cross는 학습 데이터, 굵은 실선인 ORIG는 original curve, 점선은 Bagging, dash-dot line은 Boosting, dash line은 VSG, 실선은 DVSG이다. 그림 9에서 Boosting은 과도 적합을 하고, Bagging은 약간 과소 적합을 하고 있다. 반면 VSG와 DVSG는 가상 샘플의 효과로 인해 과소 적합과 과도 적합을 하지 않는다.

표 1. 실험 결과

| Data | Error | MLP | BAG | BOOST | VSG | DVSG |
|---|---|---|---|---|---|---|
| 1 | AMSE($10^{-3}$) | 14.29 | 7.475 | 18.88 | 9.085 | 7.372 |
| | VMSE($10^{-3}$) | 1.09 | 0.002 | 0.202 | 0.011 | 0.004 |
| 2 | AMSE($10^{-3}$) | 14.34 | 9.711 | 14.92 | 9.015 | 8.235 |
| | VMSE($10^{-3}$) | 0.102 | 0.007 | 0.053 | 0.035 | 0.003 |
| 3 | AMSE($10^{-3}$) | 15.16 | 9.827 | 23.3 | 8.813 | 8.914 |
| | VMSE($10^{-3}$) | 0.087 | 0.012 | 0.321 | 0.021 | 0.032 |
| 4 | AMSE($10^{-3}$) | 164.9 | 80.8 | 143.3 | 74.87 | 90.37 |
| | VMSE($10^{-3}$) | 43.3 | 1.0 | 9.1 | 4.0 | 2.5 |
| 5 | AMSE($10^{-3}$) | 207.5 | 133.9 | 335.5 | 137.5 | 128.0 |
| | VMSE($10^{-3}$) | 13.9 | 1.5 | 429.9 | 1.1 | 1.9 |
| 6 | AMSE($10^{-3}$) | 163.6 | 114.3 | 216.0 | 122.2 | 130.5 |
| | VMSE($10^{-3}$) | 11.4 | 1.3 | 75.8 | 1.1 | 2.9 |

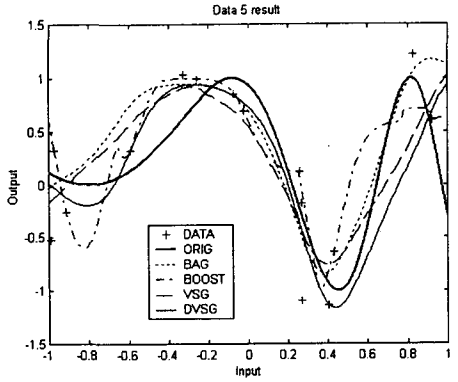


그림 9. Data 5에 대한 기법별 비교

## 6. 결론

본 논문은 VSG에 입력 분포를 고려하는 새로운 방법을 제안하였다. 이를 위해 입력 데이터의 밀도를 추정하는 과정을 정리했고, 입력 분포에 따라 적합한 가상 샘플을 생성하는 방법을 고안했다. 그러나 합성 데이터에 대한 실험 결과에 의해 VSG보다 DVSG의 성능 향상이 유의함을 입증하지는 못했다. 하지만 입력 데이터의 분포를 고려하는 것이 일반화 성능을 향상시키지 않는다고 단정할 수도 없다.

본 논문에서 제안하는 방법은 관찰 학습 알고리즘에 적용되어 비교될 수 있다. 그리고 DVSG를 보완하기 위해서는 출력 패턴을 고려하여 입력 데이터의 밀도를 추정하는 것이 보다 효과적일 것이고, 밀도가 높은 데이터에 대해서는 가상 샘플을 적게 생성하고 밀도가 낮은 데이터에 대해서는 많은 가상 샘플을 생성하는 방법도 적용될 수 있다.

* 본 연구는 뇌과학 및 공학 연구프로그램과 BrainKorea21에 의해서 지원되었다.

## 참고 문헌

[1] C. M. Bishop, " Novelty detection and neural network validation" , IEE Proc.-Vis., 1994

[2] C. M. Bishop, " Neural Network for Pattern Recogn ition" , Clarendon press, 1995

[3] H. Drucker, " Boosting Using Neural Networks " , Springer - Verlag, June 17, 1998

[4] S. Cho, M. Jang, and S. Chang, " Virtual Sample Generation Using a Population of Networks" , Neural Processing Letters, pp.83-89, 1997

[5] 권유화, 조성준, " 가상샘플 데이터를 이용한 신경망의 일반화능력 제고와 그 응용" , 정보과학회, 제25권 8호, pp.1137-1147, 1998

[6] 신현정, 장민, 조성준, 이봉기, 임용업, " 앙상블 학습 알고리즘의 일반화 성능 비교: OLA, Bagging, Boosting" , 정보과학회, 제27권 1호, pp.226-228, 2000