OSS "개발자"의 Machine Learning 분투기

머신러닝 시작부터 예측모델 배포까지

한국마이크로소프트 | 김대우

2016-11-18

SOSCON SAMSUNG OPEN SOURCE CONFERENCE

OSS "개발자"의 Machine Learning 분투기

개발자 & 머신러닝

“

뭐… 뭔데 그게?

”

deep dive: handling of time

extend our example to an RNN

$$h_1(t) = \sigma(W_1\, x(t) + H_1\, h_1(t\text{-}1) + b_1)$$
$$h_2(t) = \sigma(W_2\, h_1(t) + H_2\, h_2(t\text{-}1) + b_2)$$
$$P(t) = \text{softmax}(W_{\text{out}}\, h_2(t) + b_{\text{out}})$$
$$ce(t) = L^{\text{T}}(t) \log P(t)$$
$$\sum_{\text{corpus}} ce(t) = \max$$

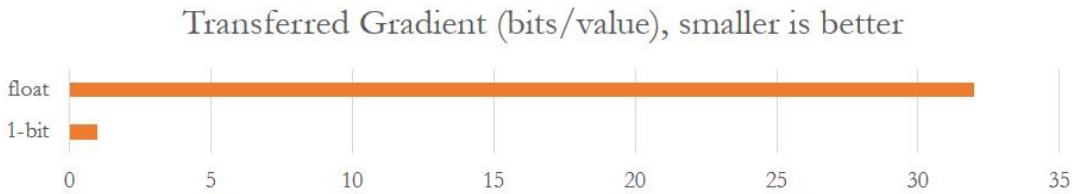
```
h1 = Sigmoid(W1 * x  + H1 * PastValue(h1) + b1)
h2 = Sigmoid(W2 * h1 + H2 * PastValue(h2) + b2)
P  = Softmax(Wout * h2 + bout)
ce = CrossEntropy(L, P)
```

→ no explicit notion of time

deep dive: 1-bit SGD

- quantize **gradients** to but **1 bit per value** with **error feedback**
  - carries over quantization error to next minibatch

$$
\begin{aligned}
G_{ij\ell}^{\text{quant}}(t) &= \mathcal{Q}(G_{ij\ell}(t) + \Delta_{ij\ell}(t-N)) \\
\Delta_{ij\ell}(t) &= G_{ij\ell}(t) - \mathcal{Q}^{-1}(G_{ij\ell}^{\text{quant}}(t))
\end{aligned}
$$



1-Bit Stochastic Gradient Descent and its Application to Data-Parallel Distributed Training of Speech DNNs, InterSpeech 2014, F. Seide, H. Fu, J. Droppo, G. Li, D. Yu

“

OK.
잠시 방황한거 뿐이야.

”


매니저
산채로 잡아라

“

어느 고수님 말씀 :

단지, 우리와 단어가 다를 뿐이야

”

“

R, SAS, Python?

”

“

Machine Learning & Cloud(?)

”

“

ML로 태어나 Cloud에서 산다

”

어디에 어떻게 사용해야 하나

머신러닝 데모

학습모델 / 예측모델

“

바보(머신)에게 공부할 기회를
= 학습모델

”

“

바보(머신)에게 배운거 물어볼까
= 예측모델

”

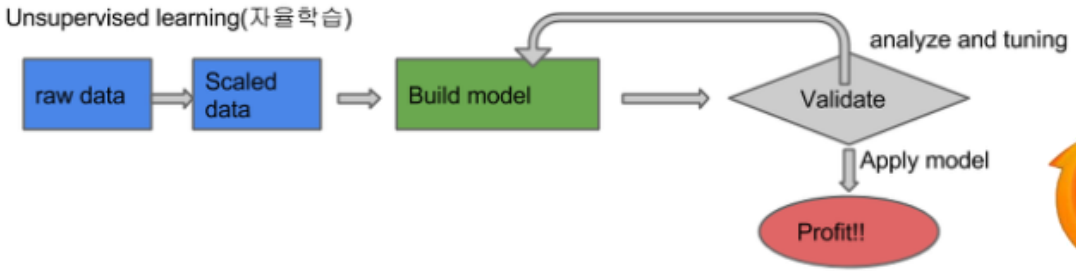
지도 학습

비지도 학습

구분	Reinforcement Learning	Machine Learning (Supervised Learning)	비고
목적함수	보상을 최대화 (또는 손실을 최소화)	오차를 최소화 (오차 = 추정 – 실제)	
산출방식	순차적으로 현재 스테이지의 보상과 총 보상을 산출하여 "총 보상"이 최대화 되도록 함	실제 사례를 기반으로 사례와 가장 유사하게 모사하도록 함수를 구성하도록 함	
산출 방법론	Optimization	분류문제와 예측문제로 구분되며 다양한 알고리즘 존재	
데이터 구성	State 별 Action Matrix (모든 가능한 State 각각에 대한 모든 실행 가능한 Action과 확률)	State vs. Action에 대한 성공과 실패 사례	
특징	규칙기반으로의 설계가 용이함 (Heuristic 설계 용이)	- 데이터마이닝: 규칙(if/else) 기반 설계 용이 - 기계학습: 규칙 파악이 어려움	기계학습은 정확도 향상이 주 목표임
구현 방법	최적화 엔진 (Dynamic LP 등)	통계 소프트웨어 또는 기계학습 엔진	

학습대상이 있는지(Target의 여부)에 따라
→ Unsupervised Learning(自律학습)과 Supervised Learning(指導학습)으로 구분

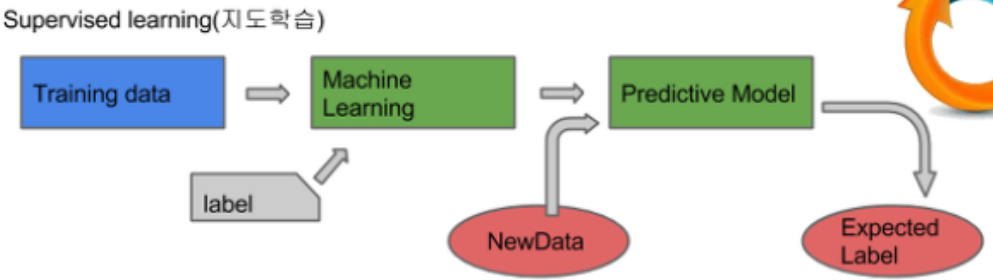
Unsupervised Learning



Clustering

- Hierarchical Clustering
- K-Means
- Mixture Modeling

Supervised Learning



Target

Continuous

- Decision Tree
- Boosting Trees
- Random Forest
- SVM(Support Vector Machine)
- Neural Networks

Discrete

- Naïve Bayes
- K-Nearest Neighbors
- Logistic Regression
- SVM

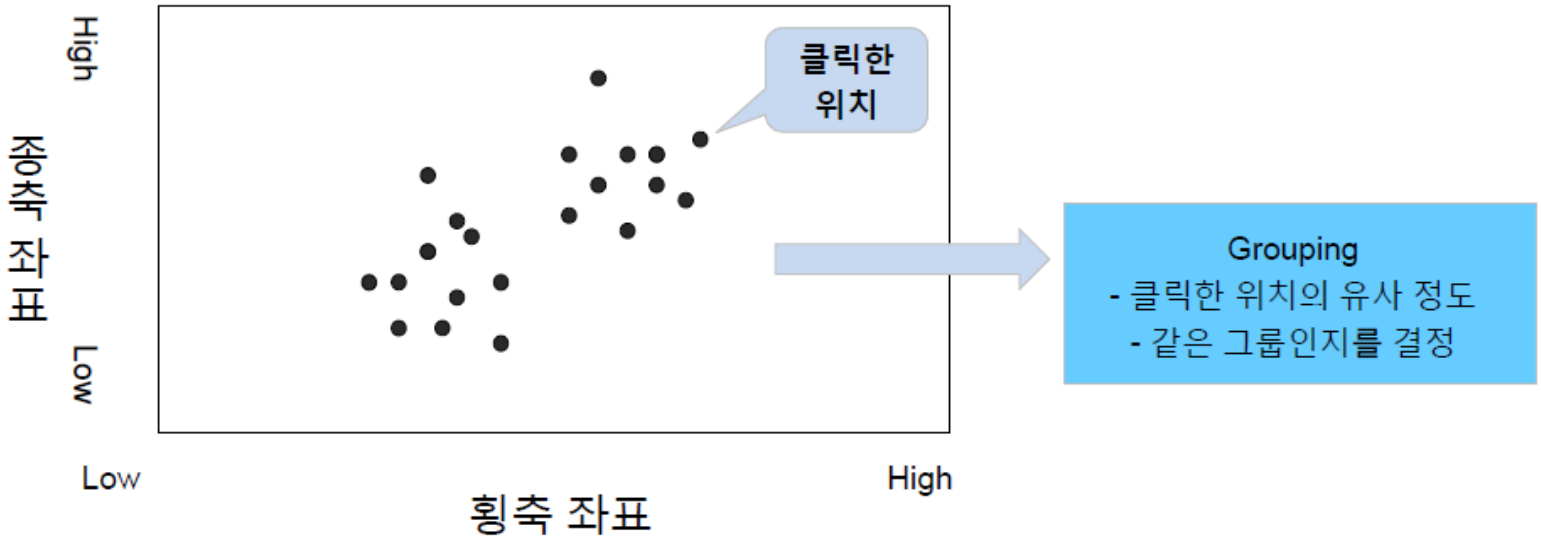
출처: http://ddiri01.tistory.com/216

다양한 분석 알고리즘

Clustering

어떤 사용자가 화면에 클릭한 위치들의 집합을 찾기 위해 **Grouping**을 한다고 하면

여기서, 각 점은 클릭한 위치를 의미



클릭한 포인트 간의 비유사도(거리)로 표현 가능) 정의

$$\text{Distance} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

$$d_{ij} = d(X_i, X_j) = [\sum_{k=1}^{p} | X_{ik} - X_{jk} |^m]^{1/m}$$

$$d_{ij} = d(X_i, X_j) = [\sum_{k=1}^{p} | \frac{X_{ik} - X_{jk}}{S_k} |^m]^{1/m}$$

군집분석 알고리즘별 원리와 특성

군집분석 알고리즘

- **Hierarchical Cluster Procedures**
  - Single Linkage Method
  - Complete Linkage Method
  - Average Linkage Method
  - Ward's Method
  - Centroid Method
- **Nonhierarchical Cluster Procedures**
  - K-means Clustering

- Agglomerative: '가까운' 객체끼리 군집화 시키는 방법
- Divisive: '먼' 객체들을 나누어 가는 방법
- 군집의 병합 또는 분리되는 과정을 이차원도면의 Dendrogram를 사용하여 간략히 표현
- 군집화 과정에서 어떤 개체가 일단 다른 군집에 속하면 다시는 다른 군집에 속하지 못함
- 개체의 수가 적을 때 유용

예측모델 생성 데모

API로 노출

Python 등에서 API 사용

Q & A

OSS "개발자"의 Machine Learning 분투기

머신러닝 시작부터 예측모델 배포까지

한국마이크로소프트 | 김대우

2016-11-18

SOSCON SAMSUNG OPEN SOURCE CONFERENCE