Relating data characteristic and algorithm and how to combine the algorithm

Selection of machine learning algorithm be based on different types of data characteristic: data size, ratio between data mean and variance.

1. Introduction

어떤 머신러닝 알고리즘을 적용할 것인지는 데이터의 특성에 따라 달라진다. 뿐만 아니라, 사용자가 어떤 종류의 결과를 원하는 지에 따라서도 달라질 수 있다. 따라서 본 논문에서는 이것을 실제 데이터를 기반으로 증명해 보이고, 각 알고리즘과 데이터의 특성간의 관계를 설명하고자 한다.

예를 들어 선형적인 데이터가 많은 곳에서는 기존 회귀 모형이 가장 적절하지만, 복잡하고 특성이 많은 데이터의 경우에는 비선형성을 표현할 수 있는 뉴럴네트워크 모형이 가장 우세하다.

또한, 데이터의 양이 적을 때는 베이지안 추론을 바탕으로 한 모형이 가장 정확하지만, 데이터가 축적되어 양이 많아질수록 뉴럴네트워크 모형이 더 정확해진다. 예를 들어 베이지안 모델을 최소 6일치의 데이터로도 다음날의 판매량을 추론할 수 있지만, 뉴럴네트워크는 불가능하다.

+변동성 계수

(어떤 특성의 결과를 원하는 지에 따라(robustness, reslilence, 이상점에 얼마나 영향을 받는 지에 따라) 알고리즘이 달라질 수 있다는 내용 추가)

따라서 산업별로 어떤 알고리즘을 적용해야 할 지가 다르고, 또 같은 산업이라도 품목에 따라 다르고, 같은 품목이라도 축적된 자료의 양에 따라 선택되는 알고리즘이 달라진다. 추가적으로 같은 산업, 같은 품목, 같은 데이터를 가지고 있다고 해도 점주가 어느 특성의 결과값을 얻고 싶어하는지에 따라 다른 알고리즘이 선택될 수 있다.

기본적으로 본 논문에서 비교한 알고리즘은 세 가지다. 이 분류법은 페드로 도밍고스가 제안한(산업컴퓨팅 개론 교과서도 유사분류 이용) 알고리즘의 5가지 분류를 기반으로 한다.

첫 번째 알고리즘은 기존에 사용되는 시계열의 수리적 기본 모형을 바탕으로 유전자 알고리즘을 통해 계수를 최적화한 모델이다.

Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering*, *129*(6), 664-672.

Chodak, G., & Kwaśnicki, W. (2000). Genetic algorithms in seasonal demand forecasting.

Chodak, Grzegorz, 2009. "[**Genetic algorithms in forecasting of Internet shops demand**](https://ideas.repec.org/p/pra/mprapa/34034.html)," [MPRA Paper](https://ideas.repec.org/s/pra/mprapa.html) 34034, University Library of Munich, Germany, revised 2009.

두 번째 알고리즘은 뉴럴 네트워크를 이용한 방법으로 각 날짜에 해당하는 판매량을 바탕으로 지도학습을 수행한다.

Law, R., & Au, N. (1999). A neural network model to forecast Japanese demand for travel to Hong Kong. *Tourism Management*, *20*(1), 89-97.

세 번째 알고리즘은 베이지안 방법이다.

West, M. (1996). *Bayesian forecasting*. John Wiley & Sons, Inc..

Harrison, P. J., & Stevens, C. F. (1971). A Bayesian approach to short-term forecasting. *Operational Research Quarterly*, 341-362.

Wong, K. K., Song, H., & Chon, K. S. (2006). Bayesian models for tourism demand forecasting. *Tourism Management*, *27*(5), 773-780.

페이스북에서 발표한 prophet이라는 demand forecasting libraray도 베이지안 방법론을 기반으로한다. 이에 대한 이론적인 설명은 다음과 같다.

페이스북에서 발표한 자료

https://facebookincubator.github.io/prophet/static/prophet\_paper\_20170113.pdf

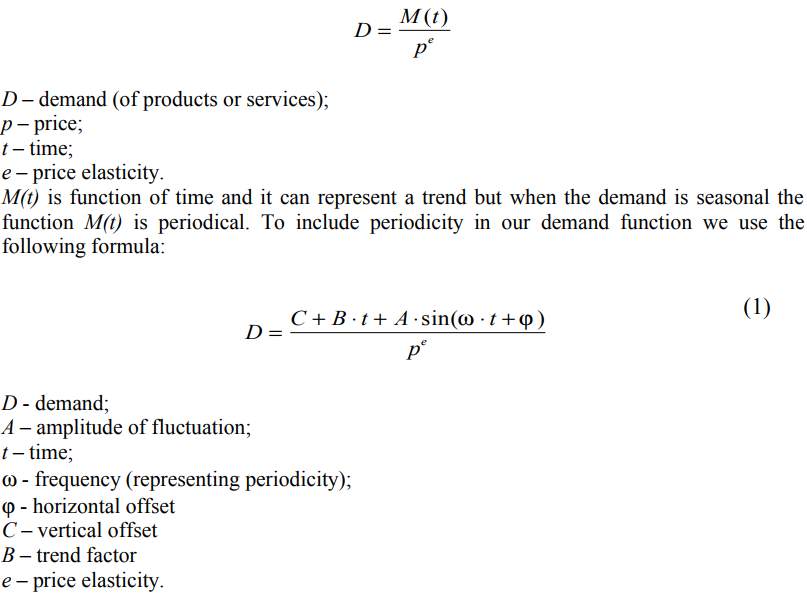
본 논문의 목적은 이 세 가지 알고리즘을 다른 특성을 가진 세 종류의 데이터에 적용시켜보고 그 결과를 분석하는 것이다. 결과를 통해 판매품목의 유형, 판매데이터의 양 등에 의해 예측오차가 작은 최적의 알고리즘이 다르다는 것을 보이고, 각 유형과 양에 따라 왜 특정 알고리즘이 다른 알고리즘보다 유리한지 설명한다. 그리고, 최종적으로 판매량을 예측해주는 시스템을 만들기 위해 이런 데이터에 따라 다른 알고리즘을 적용하는 문제를 극복하는 방식을 제안한다.

지적: 일반화하기에는 데이터 양이 너무 적다. 같은 유형별로, 같은 데이터 양별로 실험을 적어도 10개는 해야 할 듯?

그리고, api를 쓴다면 각자 최적화된 것이 다를 텐데, 이것에 대한 결과를 일방적으로 비교해도 될까? 이 문제점을 어떻게 해결할 수 있을까?

1. Methods

첫 번째 알고리즘은 기존에 일반적으로 사용하고 있는 다음 시계열 모형을 이용했다.



왜 많은 모형 중 이 모형을 선택했는지에 대한 설명추가

<http://www.ioz.pwr.wroc.pl/Pracownicy/Chodak/artykuly/genetic_algorithms_in_seasonal_demand_forecasting.pdf>

그리고 위 모형의 계수를 유전자 알고리즘을 통해 계수를 최적화했다.

두 번째 알고리즘은 뉴럴네트워크를 이용한 방법으로 tensorflow을 이용하여 RNN의 lstm을 구현하였다. 여러가지 뉴럴네트워크의 방법론 중 RNN의 lstm을 이용한 이유는 다음과 같다.

뉴럴 네트워크는 input을 output과 mapping시킬수 있다. 이때 input은 전처리가 되어있을 때 효과적이다. 그런데, 일반적인 뉴럴 네트워크는 많은 parameter와 data가 필요한데, single short time series 는 좋은 훈련을 위한 충분한 데이터를 제공하지 못 한다. 반면에 Recurrent Neural Network는 internal state를 저장할 수 있어서 short-term history에서는 같지만, long-term history에서는 다른 series에서 다르게 반응(예측)하게 학습될 수 있다. 그리고, 그 RNN 중에서 가장 좋은 good long term memory 는 lstm 알고리즘이다. An LSTM can be viewed as a powerful and complicated, nonetheless a single layer neural network, see a great introduction at http://colah.github.io/posts/2015-08-Understanding-LSTMs.

(<https://gallery.cortanaintelligence.com/Tutorial/Forecasting-Short-Time-Series-with-LSTM-Neural-Networks-2> 를 참조했음)

세 번째 알고리즘은 베이지안 방법을 토대로 하고, 이는 페이스북에서 발표한 prophet이라는 api를 이용하였다.

데이터의 경우 이 논문에서 조사하고자 하는 알고리즘 선택에 영향을 주는 특성은 여러 가지가 있겠지만, 본 논문에서는 일차적으로 데이터의 변동성을 기준으로 보았다. 그리고, 그 변동성의 대 분류 중에서 데이터의 양 또한 관찰해보았다. (이 두 특성을 선택한 이유) 따라서 test data도 이 특성들이 다르게 수집을 하였다.

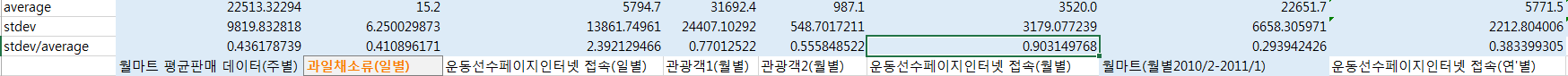
데이터가 3달, 1년치 있을 때를 비교해보았고,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Symbolist+Gen | LSTM(Connectionist) | Bayseian GAM | Best algorithm |
| 3달 데이터 |  |  |  |  |
| 1년치 데이터 |  |  |  |  |

데이터의 변동성 정도

이는 뚜레주르의 호밀샌드위치, 인터넷 옷 쇼핑몰의 드레스, 투썸플레이스의 아메리카노 세 품목의 1년 데이터를 이용했다. (관광객 데이터가 있다면 더 변동성y 할 듯)

변동성은 월별 데이터를 기준으로 하는지 주별 데이터를 기준으로 하는지에 따라서도 달라짐



이에 따르면 같은 산업체에 따른 분류는 (월별 기준)

월마트, (뚜레주르 과일채소), 인터넷 접속, 관광객1, 관광객2로 보았을 때

월마트<관광객2<관광객1<(슈퍼볼 선수 페이지) 인터넷 접속 가 되고

날짜별로는 예상대로

연별<월별<주별<일별 이 된다.

추가)

변동성이 비교적 적은

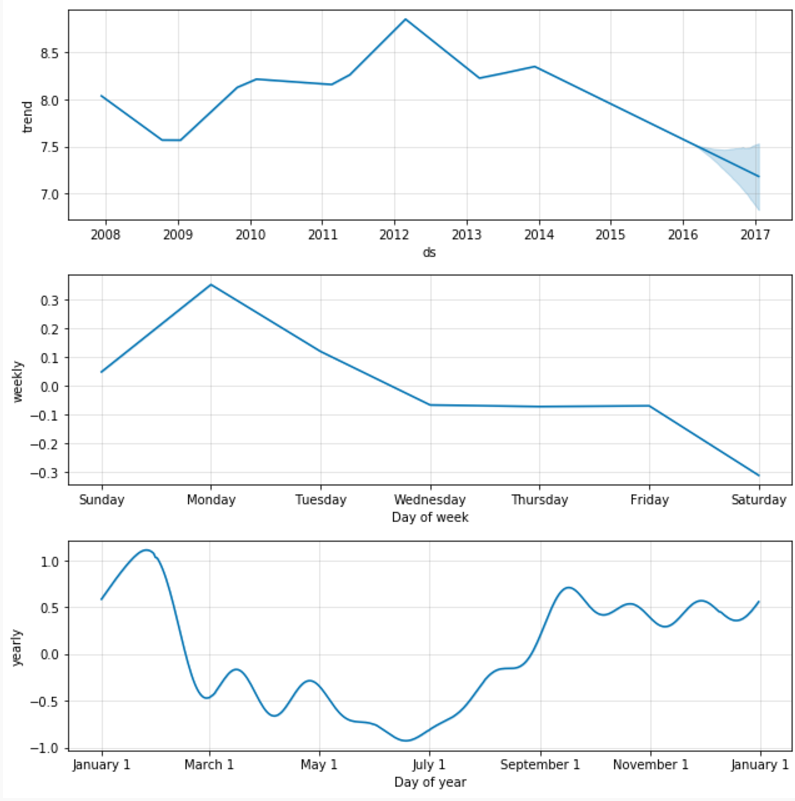
월마트의 주별데이터를 월별, 연별로 가공하기

변동성이 비교적 큰

웹페이지 접속 일별데이터를 주별, 월별, (분기별), 연별로 가공하기

어느 방식으로 분류를 할까?

* 7일을 기준으로 평균을 냈을 때 표준편차? or 7일 묶음 평균과 표준편차의 비율
* facebook의 api를 이용?....(근데, 이건 좀 객관성에 어긋나는 듯)



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Symbolist+Gen | Connectionist | Bayseian | Best algorithm |
| 변동성 작음(sd/m<?) |  |  |  |  |
| 변동성 큼(sd/m>?) |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Symbolist+Gen | Connectionist | Bayseian | Best |
| 변동성 작음 | Day<30(or 7) |  |  |  |  |
| Day>30(or 7) |  |  |  |  |
| 변동성 큼 | Day<30(or 7) |  |  |  |  |
| Day>30(or 7) |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Trend.fluct+Gene | RNN LSTM | BayesianGAM | Best |
| 변동성 작음 | Day<30(or 7) |  |  |  |  |
| Day>30(or 7) |  |  |  |  |
| 변동성 큼 | Day<30(or 7) |  |  |  |  |
| Day>30(or 7) |  |  |  |  |

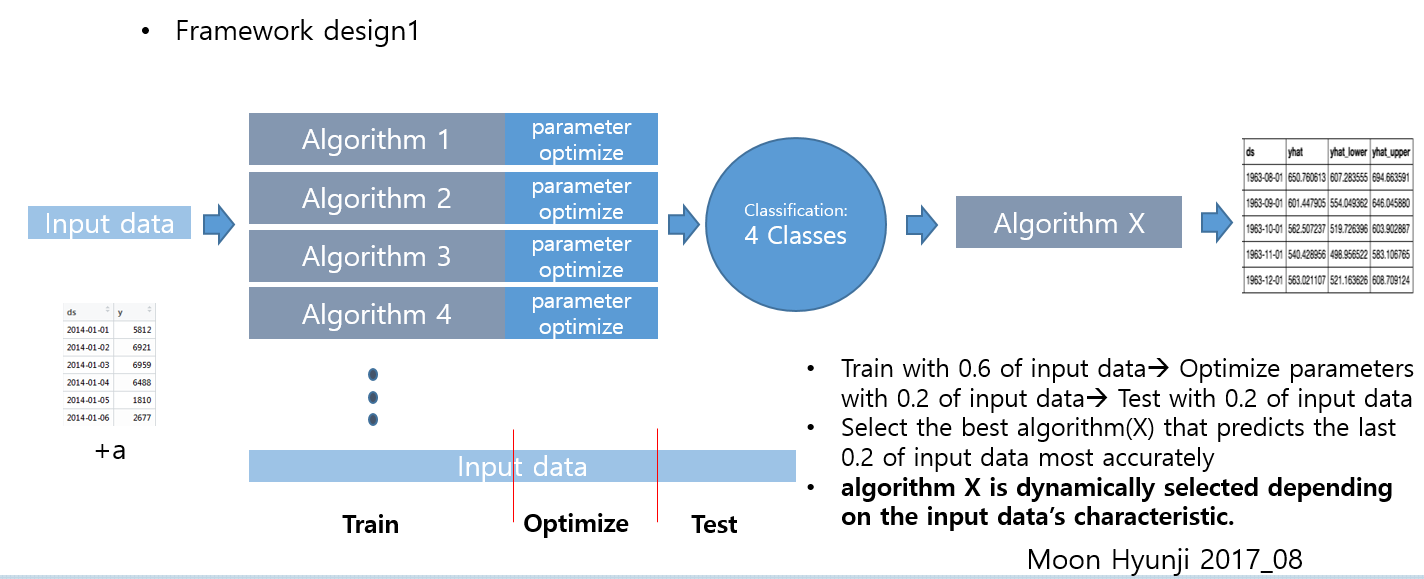
데이터의 빈도수와 자료수는 상관관계가 있다.

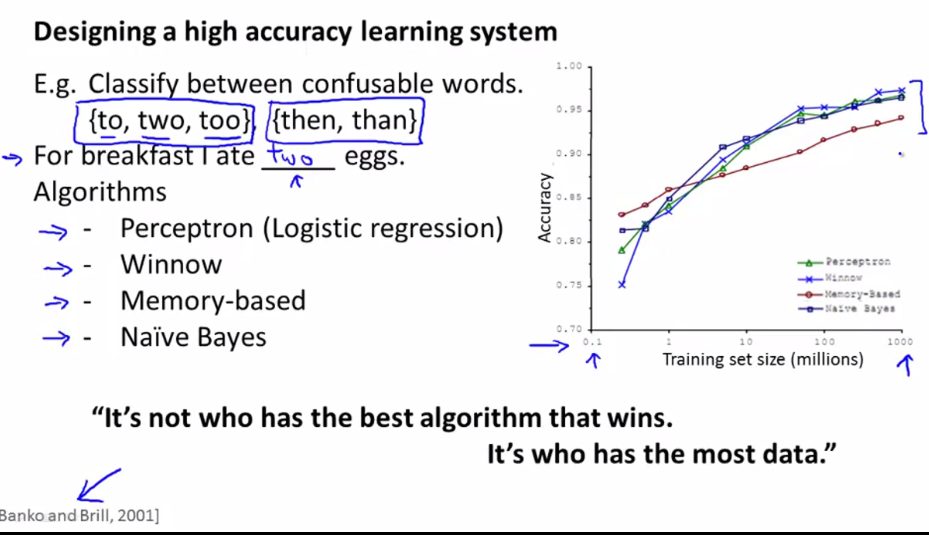
예를 들어 일별데이터는 많을 수밖에 없고, 연별데이터는 적을 수밖에 없다.

각 모델의 평가기준을 train과 test로 0.7, 0.3으로 나눈 후 test data예측값과 실제값의 rmse로 하였다.

그리고 두 번째는 데이터의 특성에 따라서 선택해야 하는 알고리즘이 달라지는 문제를 해결하고, 현실에서 사용할 수 있는 수요예측 솔루션을 위한 프레임워크를 이용하여 기존 점주들의 주문 방식과 위에서 서술한 세가지 방식, 그리고 프레임워크를 이용한 방식을 비교해보았다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 현실점주 | Symbolist+Gen | Connectionist | Bayseian | Hybrid | Best |
| 3달 데이터 |  |  |  |  |  |  |
| 1년치 데이터 |  |  |  |  |  |  |
| 3년치 데이터 |  |  |  |  |  |  |
| 변동성 작음(sd/m<?) |  |  |  |  |  |  |
| 변동성큼(sd/m>?) |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |





주기성이 있는 데이터와 주기성이 없는 데이터도 있는데

yearly, monthly, weekly avg data는 주기성이 없고

후…데이터 만드는데 정말 힘들다.

일단 wal의 달 data를 만들었고

총 데이터 수는 33개인데, 이중 0.8인 26개로 train을 하고 나머지 0.2인 7개로 test를 한다.

마찬가지로 web의 month data는

총 데이터 수가 98이고, 이 중 89개로 train을 하고 나머지 9개로 test 한다.

음. 그러면 마지막 9개를 1으로 바꾸고 입력하는데,

그러면 자기 안에서 스스로 88개의 0.2만큼을 test로 써서

**그러니까, 전체데이터의 0.1이 real test로 쓰이고,**

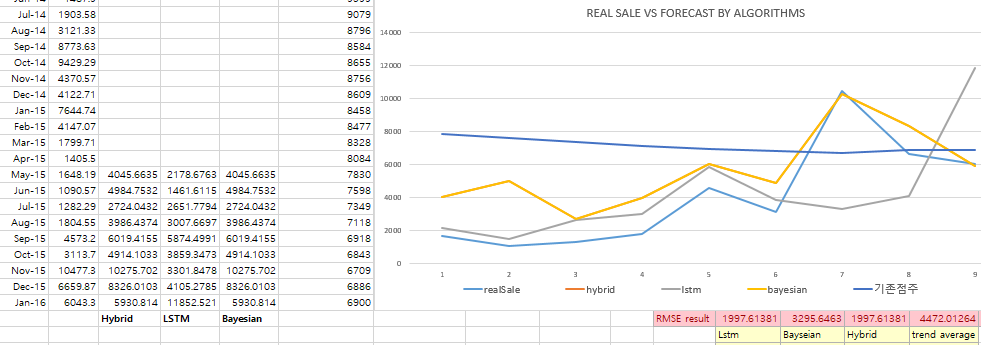
**나머지 0.2가 validation으로 쓰인다.**

내 서버의 데이터입력형식은

(알고있는 정보+ 알고싶은 정보)를 하고, 알고싶은 정보에는 0을 채워 넣어준다.

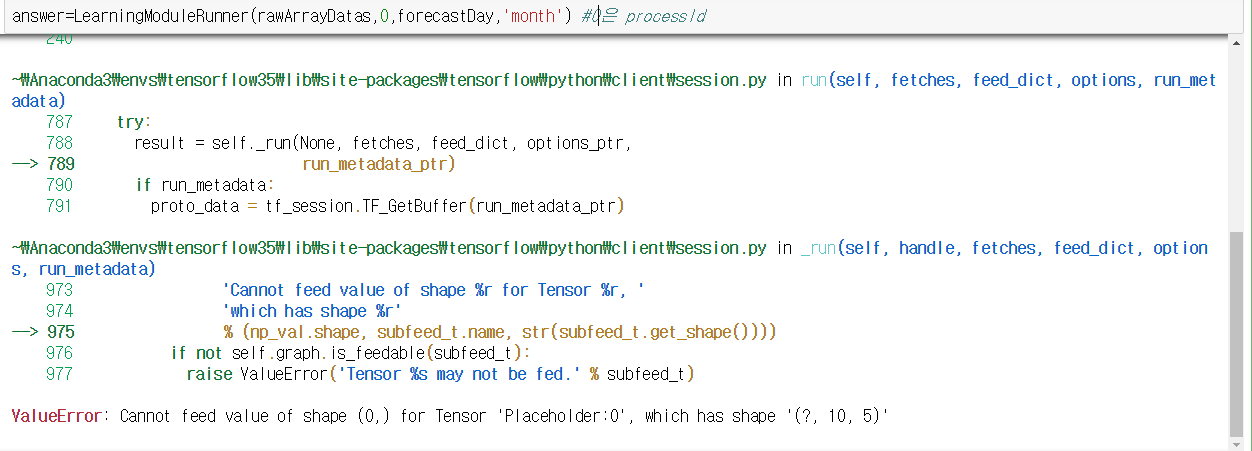
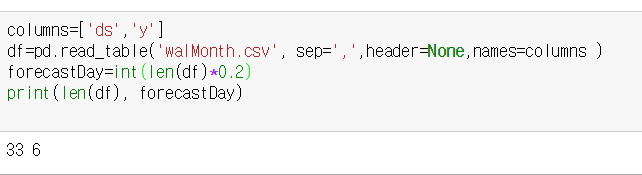
webMonth의 결과가 나왔다.

lstm 승 (변동성이 크고 데이터 양이 많을 때는 lstm이 낫다.)



몇 개 없을 때는 log를 쓰지 않는 것을 추천

이건 그냥 lstm이 이김 (log를 쓰지 않은)

* lstm은 확실히 데이터가 많이 있어야 함
* 
* 33개를 가지고 0.2인 6개를 test data 나머지 6개를 validation으로 사용하니 에러
* bayesian은 6개의 data만 있어도 예측해줌