데이터기반학습 Project 2 보고서

5조 한태구, 정유민, 문현지

1. Data Analyze

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Gold** | **Silver** | **Platinum** | **NaturalGas** | **Gasoline** |
| **Gold** | 1.000000 | 0.875931 | 0.580744 | 0.015455 | 0.662441 |
|  | **CrudeOil** | **Copper** | **BrentOil** | **AUD** | **CNY** |
| **Gold** | 0.571291 | 0.637232 | 0.684651 | 0.737895 | 0.110185 |
|  | **EUR** | **GBP** | **HKD** | **JPY** | **USD** |
| **Gold** | 0.471192 | 0.282497 | -0.198179 | 0.712148 | -0.204220 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Gold** | **Silver** | **Platinum** | **NaturalGas** | **Gasoline** |
| **Gold** | 1.000000 | 0.842140 | 0.758840 | -0.419715 | 0.034640 |
|  | **CrudeOil** | **Copper** | **BrentOil** | **AUD** | **CNY** |
| **Gold** | 0.115378 | 0.777386 | -0.265752 | 0.376456 | 0.723535 |
|  | **EUR** | **GBP** | **HKD** | **JPY** | **USD** |
| **Gold** | 0.185400 | 0.446643 | -0.749957 | -0.192880 | -0.743550 |

시간상 모든 symbol의 조합에 대해 성능을 테스트해보기 어려웠기에 통계분석을 통해 관련 없는 symbol을 제거하는 작업을 하였다. 이를 위해 위와 같이 모든 commodities와 currencies에 대해 상관분석을 2010년부터의 데이터와 2018년부터의 데이터를 가지고 각각 실행하였다. 이를 바탕으로 두 경우 모두 Gold와의 상관계수 절대값이 낮은(0.6 이하) symbol은 후보에서 제외하였고, CrudeOil과 CNY, HKD는 각각 BrentOil과 JPY, USD와 상관계수가 매우 높아 후자만을 사용하였다. 이와 같은 과정을 통해 최종적으로 'Platinum', 'Silver', 'BrentOil', 'Copper', 'AUD', 'JPY', 'USD'를 symbol의 후보로 결정하였다.

2. Data Preprocessing

(1) Normalization

스케일링의 경우 none, minmax, z-score로 시도해보았으며, 최종적으로 none이 가장 성능이 좋아 이를 선택했다.

(2) Data cleaning

각 Commodities와 Currencies별로 데이터가 존재하는 날짜가 달라 NA 데이터가 생기는데, 이를 기본적으로 linear interpolation을 사용하여 채워주었다. Linear interpolation을 사용해도 남은 NA는 Drop하였다.

3. Feature/Parameter Selection

(1) Validation

시계열에서 주로 사용되는 walk forward 방식으로 end을 고정하고, end day를 time\_step일만큼 잘라 k개의 test(validation) set을 만들어 validation error는 k개의 예측값을 평균내는 방식으로 했다. 이때 time step은 본 과제에서 최종적으로 예측해야하는 10일로 설정, k 역시 최근 trend를 잘 반영하면서도 너무 짧지 않은 10으로 설정했다.

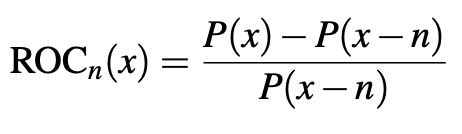
(2) Symbol Selection

1에서 추린 후보들에 Gold를 필수적으로 포함한 Symbol들의 2^7 조합에 대해 Validation한 set으로 mae를 계산한 결과 [Gold, Silver] 조합이 5.979505419672208로 제일 낮게 나왔다. 따라서 Base 조합인 [Gold]와 함께 [Gold, Silver] 조합을 symbol 선택하였다.

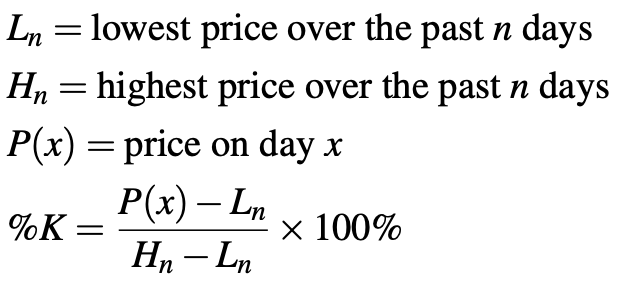
(3) Feature Selection

각 symbol에서는 price와 diff를 썼는데, diff 변화율 지표는 금일 금값과 이전 mva일 이동평균을 비교한 지표로 새롭게 정의했다. 금값 예측 문헌 중 price의 trend를 1차적으로 빼 시계열의 정상성을 확보한 후 예측을 하는 시도도 있어 구현해보았지만 유의미한 성능개선이 없었기에 제외했다.

추가적으로 금값예측에 도움이 될 수 있는 3가지 경제지표 mva, roc, so를 도입했다. mva는 n일 이동평균이며, roc는 변화율을 보기 위한 지표로 금일가격과 n일 전의 가격을 n일 전의 가격으로 나누며, so는 모멘텀 지표로 금일 가격과 이전 n일 가격의 최저가격을 n일 최고와 최저의 차이로 나눈다. Mva는 변동성 있는 price를 스무딩하여 더 robust한 feature, roc는 시계열의 추세를 반영한 feature, so는 급격하게 오르거나 내린 값을 찾기 위한 feature의 역할을 한다. 식은 다음과 같다.



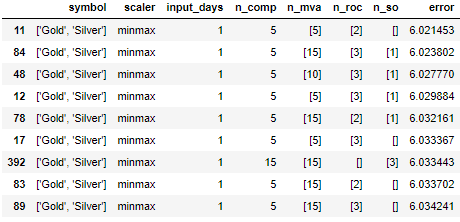
수식 1. ROC



수식 2. SO

이때 각 지표를 산출하는 n일 구간에서의 각각의 n 역시 하나의 hyper-parameter가 되므로 실험 목록에 추가했다.

(4) Parameter Tuning



위와 같은 방식으로 다양한 Hyper-Parameter 조합을 실험해본 결과, input days=[1,3,5] 중 1이 가장 좋은 성능을 보였다. n\_comp의 경우 [5,10,15]중 5인 경우가 대체적으로 성능이 좋았으나, 이보다 작은 3에 대해 따로 실험을 해본 결과 3이 더 좋은 성능을 보였다. 3개의 수정된 feature의 경우 mva = [X, 5, 10, 15], roc = [X, 1,2,3,4,5], so = [X,1,2,3,4,5]의 조합으로 실험했으며 mva, roc = [X, X] 또는 [5, 3] / so =[X]일 때가 대체적으로 좋았다.

또한 이를 기반으로 GMMHMM 모듈을 이용해 Gaussian Mixture 모델을 테스트한 결과 하나의 Gaussian만을 사용한 경우보다 더 좋은 성능을 보였다. 따라서 n\_mix(mixture 개수)도 하나의 parameter로 놓고 각각의 mixture마다 3개의 component를 가지고 있다고 하여 탐색한 결과 n\_mix가 3일 때 원래 모델에 비해 가장 개선된 성능을 보였다.

이를 기반으로 input days=1일 때를 돌려봤을 때 학습의 오류 편차가 커 input days=3,5,10,20을 같이 돌린 결과, mva, roc = [X, X]일 때는 input days=1일 때 mva, roc = [5, 3]일 때는 input days=3일 때 성능이 가장 좋았다. 따라서 input/n\_mix/n\_mva/n\_roc/n\_so = (3,3,5,3,x), (1,3,x,x,x) 총 두 종류의 Parameter Set을 최종적으로 선택하였다.

(5) Final Model

GMMHMM은 mixture의 클러스터링에 따라 결과의 편차가 커지므로, 충분히 수렴할 수 있도록 기존 n\_itern\_iter=1000로 늘렸다. 또한 위에서 선정한 두 종류의 Parameter Set으로 각각의 모델을 만들었을 때보다 두 모델을 앙상블한 경우 성능이 훨씬 안정적이었다. 이는 실제에서 멀어진 값을 다른 한 쪽이 보완해서 성능 향상에 도움을 주는 것으로 보인다. 따라서 각 Parameter Set에 대해 모델을 10개 학습한 뒤 테스트 데이터에 대한 mae값이 낮은 3개를 선정하여 총 6개의 모델을 선별, 이 6가지 모델의 predict값의 평균을 최종 predict로 선정했다.

4. Result



위의 parameter set으로 training한 HMM모델로 최종 test MAE를 계산한 결과 약 3.84가 나왔다.