

Vol Surface Manager

본 문서에서는 서울대학교 NCIA 연구실에서 한국자산평가와 함께 개발한 Vol Surface Manager에 대해 간략하게 설명하고 이를 사용하기 위한 방법을 확인할 수 있다. Vol Surface Manager는 vol surface를 보다 효율적으로 만들기 위하여 deep learning 방법을 이용하여 개발되었다.

Introduction

https://en.wikipedia.org/wiki/Volatility_smile

옵션 시장의 implied volatility는 일반적으로 고정된 것이 아니라 volatility smile이라고 불리는 특정한 패턴을 가지고 있다. 한국자산평가는 stochastic volatility model인 SABR model을 이용하여 각 옵션의 volatility smile curve를 fitting하고 있는데, 시장의 경향성을 분석하기 위하여 이를 특정 옵션 뿐만이 아닌 임의의 tenor 값으로 확장시킬 필요가 있다. 이렇게 옵션의 strike price와 tenor를 각 축으로 하여 implied volatility를 표현한 것을 implied volatility surface (vol surface)라고 부르며, 이는 각 옵션의 volatility smile로부터 만들어낼 수 있다. 하지만 시장 데이터는 sparse하며 안정적이지 못하기 때문에 단순한 interpolation으로는 안정적인 vol surface를 만들어내는 것이 어렵고, 이를 만들기 위해 전문가의 노력이 필요하다.

Vol Surface Manager의 목적은 전문가가 안정적인 vol surface를 만드는 방법을 deep learning 방법을 이용하여 학습하고 이로부터 자동적으로 vol surface를 만들어내는 것이다. 본 프로그램이 만들어낸 vol surface를 시작점으로 하여 작업하면 보다 수월하게 안정적인 vol surface를 만들어낼 수 있을 것이다.

Required Libraries

본 프로그램은 Python3 환경에서 deep learning library인 PyTorch를 이용하여 deep learning model을 구현하고 있다. 따라서 실행을 위해서는 PyTorch 라이브러리가 필요하며, GPU 환경이 지원되는 경우 이를 사용하기 위해서는 CUDA 및 CUDNN 역시 필요하다.

Directory Structure

Vol Surface Manager는 크게 `src`, `data`, `results`, `models`, `train_data`의 다섯 개의 폴더로 구성된다. 각각의 역할은 다음과 같다.

- **src**: 본 프로그램의 코드를 저장한 폴더이다. 명령어를 실행할 때에도 본 폴더 내에 들어가서 실행시켜야 한다.
- **data**: Vol surface를 만들기 위해 필요한 데이터를 넣는 폴더이다. 필요한 데이터는 market data, raw parameter data, coefficient data의 세 종류인데 이중 market data는 넣지 않아도 무방하다.
- **results**: Vol surface를 만들고 그 결과를 저장하기 위한 폴더이다. 그 외에 각종 figure들이 본 폴더에 저장된다.
- **models**: Deep learning model들을 저장하고 불러오기 위한 폴더이다.

- **train_data**: Deep learning model들을 학습시키기 위해 필요한 데이터를 넣는 폴더이다. 필요한 데이터는 market data, raw parameter data, coefficient data의 세 종류인데 이중 market data는 넣지 않아도 무방하다. Candidate model을 학습시키는 경우 데이터 수집이 오래 걸리므로 수집한 데이터를 본 폴더에 자동 저장시킨다.

프로그램을 실행할 때 다음과 같은 몇가지 옵션을 추가함으로써 사용하는 폴더의 이름이나 경로를 변경할 수 있다.

- **DATA_BASE_DIR**: data 폴더 대신 다른 경로를 사용한다.
- **RESULT_BASE_DIR**: results 폴더 대신 다른 경로를 사용한다.
- **MODEL_BASE_DIR**: models 폴더 대신 다른 경로를 사용한다.
- **TRAIN_DATA_BASE_DIR**: train_data 폴더 대신 다른 경로를 사용한다.

Required Data

Vol Surface Manager를 사용하기 위해 **data** 혹은 **train_data** 폴더에 넣어야 하는 데이터는 market data, raw parameter data, coefficient data의 세 종류이다. 각 데이터가 의미하는 바는 다음과 같다.

- Market data는 옵션의 spot price, strike price, tenor, implied volatility 등 각종 정보들을 가지고 있다.
- Raw parameter data는 옵션의 volatility smile curve로부터 fitting한 SABR parameter 들의 정보를 가지고 있다.
- Coefficient data는 만들어진 vol surface를 표현하기 위해 각 SABR parameter에 대한 fitting curve들의 정보를 가지고 있다.

프로그램이 원활하게 데이터들을 불러오기 위해 각 데이터의 이름은 특정한 형태여야 하는데 이는 다음과 같다.

- Market data의 경우 지수명 뒤에 "_Data.xlsx"를 붙인 이름이어야 한다. 이는 smile curve 를 보기 위한 데이터이므로 없어도 무방하다.
- Raw parameter data의 경우 지수명 뒤에 "_Parameter.xlsx"를 붙인 이름이어야 한다.
- Coefficient data의 경우 지수명 뒤에 "_coef_Data.xlsx"를 붙인 이름이어야 한다.

프로그램을 실행할 때 다음과 같은 몇가지 옵션을 추가함으로써 사용하는 마켓 데이터의 이름 형태를 변경할 수 있다.

- **MARKET_DATA_SUBNAME**: Market data의 이름 형태를 지수명과 MARKET_DATA_SUBNAME를 합친 형태로 사용한다. 기본값은 "_Data"이다.
- **RAW_PARAM_DATA_SUBNAME**: Raw parameter data의 이름 형태를 지수명과 RAW_PARAM_SUBNAME를 합친 형태로 사용한다. 기본값은 "_Parameter"이다.
- **COEF_DATA_SUBNAME**: Coefficient data의 이름 형태를 지수명과 COEF_DATA_SUBNAME를 합친 형태로 사용한다. 기본값은 "_coef_Data"이다.

Methods

오늘자 vol surface를 만들기 위해서는 α, ρ, ν 의 각 SABR parameter들에 대해 tenor를 x 축으로 하여 curve fitting을 할 필요가 있다. Vol Surface Manager에서는 이를 위해 우선 오늘자 raw

parameter에 해당하는 point들과 어제 parameter curve 위에서 기준만기에 해당하는 point들 중 특정한 point들을 선택한 뒤 이들을 이용하여 fitting이 이루어지게 된다. 즉 안정적인 vol surface를 만들기 위해 적절한 point들을 선택하는 것이 중요하다.

Vol Surface Manager는 크게 Loss Model과 Candidate Model로 구성된다. 이 절에서는 Loss Model과 Candidate Model이 어떤 것인지 간략하게 설명하고, 본 프로그램이 이들로부터 어떻게 vol surface를 만들어내는지 방법도 기술하였다.

1. Loss Model

Loss Model은 vol surface를 만들기 위한 가장 핵심적인 deep learning model이다. 이 model은 현재 선택된 point들의 정보로부터 구성된 curve가 얼마나 좋은 curve인지를 수치로 표현한다. 이때 학습은 한국자산평가의 전문가들이 만들어낸 curve와의 loss를 이용하여 이루어지게 된다.

2. Candidate Model

Loss Model만으로 vol surface를 예측하려면 모든 선택할수 있는 point들의 조합을 전부 loss model에 넣어봐야 하는데 이는 너무 많은 연산량을 요구하므로 이를 줄이기 위한 deep learning model인 Candidate Model이 도입되었다. Candidate Model은 각각의 점들이 curve fitting에 얼마나 좋은 영향을 줄지를 개략적으로 알려주며, 이는 loss model만큼 정밀하지는 못하지만 loss model이 선택할 point들의 후보군을 줄여주는 역할을 한다. 이 model 역시 학습은 한국자산평가의 전문가들이 만들어낸 curve와의 loss를 이용하여 이루어지게 된다.

3. Main Algorithm

이제 vol surface를 만드는 알고리즘은 Loss Model과 Candidate Model을 혼합하여 이루어지게 된다. 우선 전체 사용가능한 point들 중 Candidate Model이 가장 사용해야 할 것으로 보이는 point들을 선택해낸다. 이제 선택된 point들 내에서 각 조합마다 Loss Model이 얼마나 적절한 curve가 만들어졌는지를 판단하며, 가장 좋은 수치를 낸 curve가 선택된다. 이 작업은 α , ρ , ν 각각에 대해 이루어지며, 각 SABR parameter들에 대해 curve fitting이 이루어지면 최종적으로 vol surface가 완성된다.

Usage

Vol Surface Manager는 두개의 메인 프로그램으로 구성되며, 하나는 deep learning model을 학습시키기 위한 프로그램이고 또 하나는 학습된 model로부터 vol surface를 만들기 위한 프로그램이다. 이 절에는 각각을 어떻게 사용하는지 상세한 설명을 기술하였다.

1. Train the Model

Vol Surface Manager를 학습시키기 위하여 우선 train_data 폴더 내에 학습시키기 위한 data들을 넣어준 뒤에 src 폴더 내에서 다음과 같은 명령어를 실행하면 된다.

```
python kap_runme_train.py --DATA_TYPE="Kospi200" --TRAIN_MODEL_TYPE="Loss"
--TRAIN_PARAMETER_TYPE="Alpha"
```

각각의 옵션이 의미하는 바는 다음과 같다.

- **DATA_TYPE**: 어떤 지수를 다룰지 선택한다. 현재는 “Kospi200”, “S&P500”, “Eurostoxx50”, “HSCEI”를 지원하고 있다.
- **TRAIN_MODEL_TYPE**: Loss Model과 Candidate Model 중 어떤 model을 학습시킬지 선택한다. “Loss”와 “Candidate” 중 설정하면 되며 최종적으로 두 model들을 전부 학습시켜야 한다.
- **TRAIN_PARAMETER_TYPE**: 어떤 SABR parameter에 대한 model을 학습시킬지 선택한다. “Alpha”, “Rho”, “Nu” 중 설정하면 되며 최종적으로 세 종류의 parameter들에 대한 model들을 전부 학습시켜야 한다. 즉 위의 TRAIN_MODEL_TYPE 옵션과 함께 한 지수당 총 6개의 model들을 학습시키게 된다.
- **SAVE_MODELS** (optional): Model이 학습되는 동안 주기적으로 그 변화를 저장해줄지 여부를 설정한다. 이 옵션을 False로 설정하는 경우 학습 도중 가장 validation loss가 적은 model만 model_best.th의 이름으로 저장되며 중간 model들은 따로 저장되지 않는다. True, False 중 선택하면 되며 따로 설정하지 않는 경우 True로 설정된다.
- **LOG** (optional): 어떤 수준의 메시지부터 볼지 설정한다. 옵션은 “DEBUG”, “INFO”, “ERROR”이며 기본적으로는 “INFO”를 사용한다.

학습의 내용은 주기적으로 logger가 출력하므로 이를 통해 학습이 원활하게 이루어지고 있는지 여부를 확인할 수 있다.

2. Create new Vol Surface

Vol Surface Manager를 이용하여 특정 날짜, 특정 지수에 대해 새로운 vol surface를 만들어내기 위하여 우선 data 폴더 내에 그 날에 해당하는 data들을 넣어준 뒤에 src 폴더 내에서 다음과 같은 명령어를 실행하면 된다.

```
python kap_runme_vol_surface.py --DATA_TYPE="Kospi200" --VOL_DATE="20190903"
```

각각의 옵션이 의미하는 바는 다음과 같다.

- **DATA_TYPE**: 어떤 지수를 다룰지 선택한다. 현재는 “Kospi200”, “S&P500”, “Eurostoxx50”, “HSCEI”를 지원하고 있다.
- **VOL_DATE**: 어떤 날짜에 대한 vol surface를 만들지 선택한다. 옵션을 따로 설정하지 않는 경우 날짜는 오늘로 자동 설정된다.
- **PLOT_FIGURES** (optional): Vol surface를 만드는 도중 작업을 visualize하기 위한 figure들을 출력할지 여부를 설정한다. True, False 중 선택하면 되며 따로 설정하지 않는 경우 True로 설정된다.
- **SAVE_FIGURES** (optional): Vol surface를 만드는 도중 작업을 visualize하기 위한 figure들을 results 폴더에 자동저장할지 여부를 설정한다. True, False 중 선택하면 되며 따로 설정하지 않는 경우 False로 설정된다.
- **LOG** (optional): 어떤 수준의 메시지부터 볼지 설정한다. 옵션은 “DEBUG”, “INFO”, “ERROR”이며 기본적으로는 “INFO”를 사용한다.

이때 코드가 원활하게 돌아가기 위해서는 vol surface를 계산하는 날과 그 전날의 raw parameter data가 필요하며 전날의 coefficient data 역시 필요하다.

프로그램이 vol surface를 만들어낸 이후에 최종적으로 **results** 폴더에 저장하는 파일은 크게 두 종류이다. 사용자는 이 두 종류의 파일과 작업 도중 보여진 figure들을 이용하여 프로그램이 만든 vol surface를 확인할 수 있을 것이다.

- predicted coefficient data는 만들어진 vol surface에 대응하는 coefficient data를 포함한다. 이는 지수명 뒤에 "_coef_Data_Prediction.xlsx"를 붙인 이름으로 저장된다.
- predicted selection data는 만들어진 vol surface가 어떤 point들을 사용하여 구성되었는지 자세한 사항을 포함한다. 이는 지수명 뒤에 "_coef_Data_Selection.xlsx"를 붙인 이름으로 저장된다.

Jupyter Support

Vol surface를 만드는 부분을 python code를 실행하는 방법 외에 jupyter notebook을 이용하여 인터넷 상으로도 돌릴 수 있도록 하였다. `kap_runme_vol_surface.ipynb` 파일을 jupyter notebook 상에서 실행시키면 앞서 vol surface를 만들기 위한 명령어를 실행하는 것과 동일한 작업을 jupyter notebook 상에서 확인할 수 있다.