CS229 Project Milestone Calibrate Time Series by LSTM

Jencir Lee jli14

Introduction

The Econometrics has established some standard time series models: Autoregressive Integrated Moving Average (ARIMA) model [Hamil1994], Regime-Switching model [Hamil1994], Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model [Engle1982] [Boll1986], to name a few, for modelling macroeconomy, finanical markets, or general time series. The calibration of these models is generally tailored to the structure of each model, and for most of them, it usually involved conducting the calibration by stages; at each stage, a specific theory would establish the consistency and convergency guarantees of the estimator for the parameters being estimated.

In this study with limited scope, we propose to apply the Long Short-Term Memory (LSTM) model [Hoch1997] to the calibration of a few typical time series models. We'd pre-fix the parameters of the time series model, simulate trajectories from it, so that we would be able to understand the prediction performance in relation to the ground truth. The second fold of this study's objective, would be to see if the internal units (Hidden Cell, Memory Cell) of the LSTM network actually captured statistical information about the latent states of the time series models, for complex ones that do involve latent variables.

The LSTM Model

Architecture We use the peepholed-version of LSTM, so that the evolution is driven by the Memory Cell c_t and the observations x_t and effectuates on c_t, t being the time step. The hidden cell h_t would be a by-product of non-linear transformation from c_t, x_t and used to form the objective function. We argue that this architecture makes more sense econometrically as c_t solely encodes the context and h_t is for the task at hand.

Loss Function Given input series x_t, at time t, we perform a linear projection $g(x_t, h_t)$ to forecast x_{t+1}. In this way we subsume linear models. We use two metrics for training: RMSE on the forecast error $\delta_t = x_{t+1} - g(x_t, h_t)$, and Quantile Loss for the 50%-quantile of x_{t+1},

which is to minimise $\delta_t \left(\pi - \mathbb{1}_{\{\delta_t<0\}}\right)$, where $\pi = 0.5$. For evaluation, we only report the RMSE on the test set. The first b values on every time series are for "burn-in" and disregarded for optimisation, evaluation, or statistical test.

Training We use the LSTMCell built in TensorFlow. It's as simple as specifying the number of units, and input/output keep probabilities. We train with SGD with Momentum, the momentum parameter is 0.5. This has equal performance as more complex algorithm. The step size slowly decreases across iterations.

An ARIMA(2,0,2) Process

We first took the S&P 500 end-of-day price series since 1980, computes its log return, and fit a best ARIMA model to it. The selected model was ARIMA(2,0,2), Below are the parameters information and their standard errors. From the variance σ^2 of the innovation term we could compute $\sigma = 0.0112$.

```
Coefficients:
         ar1     ar2      ma1      ma2  intercept
      0.0868  0.3667  -0.1150  -0.4068      3e-04
s.e.  0.1636  0.1446   0.1603   0.1427      1e-04

sigma^2 estimated as 0.0001248:  log likelihood = 28662.03,  aic = -57312.06
```

We then simulate 5000 trajectories with 1000 time steps using these parameters for training, and 500 such trajectories for test. The burn-in period $b = 50$. If we denote one simulated trajectory by $\{y_t\}$, the innovation terms $\{\epsilon_t\}$, $t = 1, \ldots, 1000$, as we know all their values, we'd readily know the expectation $\hat{y}_t = E(y_t|\mathcal{F}_t)$ as of t, where $\mathcal{F}_t$ denotes all the information immediately before t. We could study our forecast error w.r.t $\{\hat{y}_t\}$, denoted by $\{\zeta_t\}$, $t = 1, \ldots, 1000$.

We also replace the Normal innovation distribution by a $t-$Distribution with $df = 4.5$ to generate extra kurtosis, and by an Exponential Distribution to generate asymmetry for experiments. Both would be shifted and scaled when necessary to have zero mean and the same deviation as the Normal innovation distribution.

Relation between $\{\zeta_t\}$ and $\{\epsilon_t\}$

We could perform Spearman rank-based correlation between the forecast error $\{\zeta_t\}$ and $\{\epsilon_t\}$, and compute the p-value of the null for zero correlation. The p-value was always 0.3-0.8 for Normal innovation terms for the two loss functions, which means our forecast error is

always *orthogonal* to the generative noise. When the innovation follows the $t-$ Distribution this behaviour holds; however for the asymmetric Exponential Distribution the p-value decreases to 0.07.

This points to the necessity of performing model averaging, aided by the fact that both loss functions are convex w.r.t the prediction variable.

Model Averaging

We did bootstrap study and found for both losses it usually takes 20 independent runs to produce an averaged model that could achieve a stable RMSE on the test set 0.004-0.005 (37%-44% σ). For non-Normal innovation distributions, the test RMSE of the averaged model is comparable to that for the Normal distribution.

When we increase further the number of independent runs, the RMSE on the test set wouldn't reduce proportionally, for the reason that although the forecast errors are orthogonal to the true innovation terms, there is non-zero correlation across the forecast error series empirically.

Dropout Probability

So we obtain an averaged model with 20 independent runs, for input/output keep probability 0.7, 0.8, 0.9, 1.0. They would all produce the same order of RMSE on the test set 0.004-0.005. We therefore don't apply Dropout from now on. One possible explanation is that unlike vision, the end-of-day price series is already sparse enough not to disregard any features.

Consistency of Estimator

We test the null hypothesis that the median of the forecast errors is zero with the Wilcoxon signed rank test and list the p-value below.

Innov. Dist	RMSE Loss	Quantile Loss
Normal	0	0.35
$t_{4.5}$	0.005	0.011
Exponential	0	0

We could see when the innovation is Normal, Quantile Regression would produce a consistent estimator. This ability will be dampened when the kurtosis of the innovation increases. It's understandable that for Exponential distribution the p-value is zero for the Quantile Regression as the Exponential distribution is asymmetric and the median wouldn't be at zero.

A "Bayesian Fitting"?

Doesn't the averaged model behave like performing Bayesian shrinkage? We did an interesting experiment. Now we use Stan, a Probabilistic Programming software, and specified a Bayesian version of the ARIMA(2,0,2) model, i.e with AR, MA parameters taken from zero-centered Normal prior distribution, and σ from an Inverse Gamma prior distribution.

To great surprise, although the generative process is linear, the MCMC mixing is very hard, if ever possible even after 2000 samplings, however stringent or relaxed the priors are. The posterior of the parameters would be multi-modal. We hand-picked some combination of the parameters from the posterior and the best RMSE on the test set was 0.0058 (52% σ), much in line with the fitting result by LSTM.

The conjecture is: given that the LSTM's weights are by default initialised from zero-centered Normal distributions, is this implicitly a Bayesian fitting?

References

- [Hamil1994] James D. Hamilton, *Time Series Analysis*, 1994.
- [Engle1982] Robert F. Engle, *Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation*, Econometrica, 50 (4), 1982.
- [Boll1986] Tim Bollerslev, *Generalized Autoregressive Conditional Heteroscedasticity*, Econometrics, 31 (3), 1986.
- [Hoch1997] Sepp Hochreiter, Jürgen Schmidhuber, *Long short-term memory*, Neural Computation, 9 (8), 1997.
- [Vill2009] Cédric Villani, *Optimal Transport: Old and New*, Grundlehren der mathematischen Wissenschaften, 2009.