Adnan Contractor

Contents

1 Bid Adjustment

Focus areas:

1. Accurate classification of market states to adjust trading strategies.

2. Real-time prediction of market liquidity to optimize bid placements.

3. Development of optimal execution strategies that minimize market impact.

I want to understand if we can leverage some statistical algorithms - potentially autoregressive integrated moving average ARIMA or Gaussian mixture models GMMs to model enhanced liquidity seeking capabilities.

2 Liquidity Prediction

Apply ARIMA models to predict future liquidity states based on historical liquidity data. ARIMA at a high level captures time-dependent structure in liquidity data streams. The model is realized as a function $ARIMA(p, d, q)$ with p order of its autoregressive component, d degree of differencing, and q order of its moving average component. Given a time series $\{X_t\}_t$ one has

$$X_t = c + \left(\phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p}\right) + \left(\epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}\right) \tag{1}$$

c is an error corrective constant, ϕ_i are autoregressive parameters and θ_i are moving average parameters, and $\epsilon_t \sim \mathcal{N}(\mu = 0, \sigma)$ are i.i.d that simulate noise.

Atomically ARIMA is specified by an autoregression

$$AR(p) : \left(1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p\right) X_t = c + \epsilon_t \tag{2}$$

Where B is backshift. This is intended to model the influence of past liquidity on future liquidity states. We also difference the time series to achieve stationarity

$$\Delta^d X_t = X_t - X_{t-1} \tag{3}$$

And a moving average

$$MA(q) : X_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} \tag{4}$$

Which is a modulator of anomalies in liquidity streams. There are three parameters p, d, q which we determine using (partial) autocorrelation functions, which measure the correlation between steps of a time series $\{X_t\}_t$ with mean $\overline{X}$ and lag k

$$ACF(k) = \frac{\sum_{t=k+1}^{N} (X_t - \overline{X})(X_{t-k} - \overline{X})}{\sum_{t=1}^{N} (X_t - \overline{X})^2} \tag{5}$$

d is determined by looking at the stationarity. If the ACF shows a slower decay the series is more non-stationary. In this case we'd put $d > 0$. We want to find the minimal d such that the differenced series achieves stationarity, iteratively updating $X_t \to Y_t$ according to backshift

$$Y_t = \Delta^d X_t = (1 - B)^d X_t, \; BX_t \equiv X_{t-1} \tag{6}$$

q is determined by the ACF plot: if we have strong spikes we can measure the order of the moving average by taking the cut off gap (if ACF cuts off after n lags we'll put $q = n$).

We determine p by considering the partial ACF

$$PACF(k) = \mathrm{corr}(X_t, X_{t-k} | X_{t-1}, ..., X_{t-k+1}) \tag{7}$$

p is approximated by the cutoff gap in the PACF plot.

Post parameter estimation we conduct a small residual analysis to check that residuals are first order approx. by white noise. For example use Ljung-Box to compute autocorrelation with N observations, m test lags and sample autocorrelation $\hat{\rho}_k$

$$Q = N(N+2) \sum_{k=1}^{m} \frac{\hat{\rho}_k^2}{N-k} \tag{8}$$

We can then train ARIMA on some preferably clean liquidity data stream to forecast short and medium-term liquidity flows.

3 Liquidity Classification

Implement GMMs to classify market states based on real-time order book data, and identify different liquidity regimes (e.g., high liquidity, low liquidity, high volatility). GMMs realize the distribution of market liquidity states by assuming that data is sampled from a Gaussian mixture. Given k Gaussian components and $\{\phi_i\}_i$ mixing parameter set we have probability density

$$p(x) = \sum_{i=1}^{k} \phi_i \mathcal{N}(x|\mu_i, \Sigma_i) \tag{9}$$

Σ_i is a covariance matrix entry, and

$$\mathcal{N}(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\bigg(-\frac{1}{2}(X_i - \mu_j)^T \Sigma_j^{-1}(X_i - \mu_j)\bigg) \tag{10}$$

The difficult part is parameter estimation, but one place to start is an exp-max algorithm, which initially computes an expectation or probability that some data point belongs in a Gaussian component and then maximizes that expected log-likelihood. Given initial chosen values ϕ_i, μ_i, Σ_i, the expectation computes responsibilities γ_{ij}

$$\gamma_{ij} = \frac{\phi_j \mathcal{N}(x_i|\mu_j, \Sigma_j)}{\Sigma_{k=1}^{K} \phi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)} \tag{11}$$

the maximization step (update step) optimizes

$$\log L = \sum_{i=1}^{N} \log\bigg(\sum_{j=1}^{k} \phi_j \mathcal{N}(x_i, \mu_j, \Sigma_j)\bigg) \tag{12}$$

over an (iteratively updated) set of parameters

$$\mu_j = \frac{\sum_{i=1}^{N} \gamma_{ij} X_i}{\sum_{i=1}^{N} \gamma_{ij}},\ \Sigma_j = \frac{\sum_{i=1}^{N} \gamma_{ij}(X_i - \mu_j)(X_i - \mu_j)^T}{\sum_{i=1}^{N} \gamma_{ij}},\ \phi_j = \frac{\sum_{i=1}^{N} \gamma_{ij}}{N} \tag{13}$$

With normalization constraint

$$\sum_{j=1}^{k} \gamma_{ij} = 1 \tag{14}$$

4 Bid Adjustment

At a high level the ARIMA model forecasts short-term liquidity while GMMs classify market conditions into different states. I am interested in exploring if by integrating these models an ATS can optimize bid placements based on both predicted liquidity levels and classified market states.

As a sketch, structure a feature vector X_t with order book depth, trade volumes, price differentials, perhaps other (?) components and liquidity state L_t. Given initialized ARIMA parameter set $\theta = \{\phi_i, \theta_j, d = \sigma^2\}$ define the likelihood of observing $\{L_t\}_{t=1}^{T}$ by

$$L(\{\theta\}|\{L_t\}_{t=1}^{T}) = \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\bigg(-\frac{(L_t - \mu_t)^2}{2\sigma^2}\bigg) \tag{15}$$

Giving log-likelihood

$$\log L(\{\theta\}|\{L_t\}_{t=1}^T) = -\frac{T}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{T}(L_t - \mu_t)^2 \tag{16}$$

With

$$\mu_t = c + \phi_1 L_{t-1} + \phi_2 L_{t-2} + \cdots + \phi_p L_{t-p} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} \tag{17}$$

Update the parameter set $\{\theta\} \to \{\theta\}'$ per

$$\{\theta\}' = \{\theta\} - \left[\frac{\partial^2 \log L(\theta)}{\partial\theta_i \partial\theta_j}\right] \nabla \log L(\{\theta\}) \tag{18}$$

We forecast L_{t+h} conditioned on $\{L_t, L_{t-1}, ..., L_{t-p}\}$:

$$\hat{\theta}_{\text{ARIMA}} = \arg\ \max_\theta \sum_{t=1}^{T} \log \mathbb{P}(L_t|L_{t-1}, \cdots, L_{t-p}; \{\theta\}) \tag{19}$$

We take

$$\hat{L}_{t+h} = \mathbb{E}[L_{t+h}|L_t, \cdots, L_{t-p}; \hat{\theta}_{\text{ARIMA}}] \tag{20}$$

Here μ is the series mean. In this example it reduces to

$$\hat{L}_{t+h} = \mu + \sum_{i=1}^{p} \phi_i X_{t+h-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t+h-j} \tag{21}$$

For the market condition classification bit, assume the liquidity states are generated from a mixture of k Gaussians. Fit the corresponding GMM to $\{X_t\}_t$ per

$$\hat{\theta}_{\text{GMM}} = \arg\max_\theta \sum_{t=1}^{T} \log\left(\sum_{i=1}^{k} \phi_i \mathcal{N}(X_t|\mu_i, \Sigma_i)\right) \tag{22}$$

Here $\theta = \{\phi_i, \mu_i, \Sigma_i\}_{i=1}^k$ is the GMM parameter set. Then classify a market state X_t into a Gaussian component

$$S_t = \arg\max_i \phi_i \mathcal{N}(X_t|\mu_i, \Sigma_i) \tag{23}$$

With accuracy metric

$$\Delta = \frac{1}{T}\sum_{t=1}^{T} \chi(S_t = L_t) \tag{24}$$

Here χ is the indicator. This is meant to monitor the classification accuracy.

So our final steps are to (1) integrate the ARIMA forecasts and GMM classifications into an ATS bidding engine - bids are adjusted in real-time based on the predicted liquidity levels and market states, and (2) to estimate sensitivity constant sets $\{\alpha_{L_t}\}_t$ and $\{\alpha_{S_t}\}_t$:

$$B_t = f(\hat{L}_{t+h}, S_t) = P_t + \alpha_{L_t} \cdot \hat{L}_{t+h} + \alpha_{S_t} + \sigma \tag{25}$$

which is our functional representation for the bid adjustment B_t based on (1) a liquidity forecast and (2) a market classification up to error σ.

An example implementation strategy might look like

1. Collect real time data on book depth, volumes, spreads, volatility, etc. to populate a feature vector X_t
2. Use trained GMM to classify corresponding state S_t
3. Determine $\hat{L}_{t+h}$ using ARIMA(p, q, d) with p, d, q estimated by (P)ACF
4. Adjust B_t per above
5. Validate functional expression per error term σ