Team “The Rats”

UMHackathon 2025 : Balaena Quant Problem Statement

A Multi-Factor Framework for Alpha Generation in Crypto Markets
Integrating Structural Analysis, Regime Detection, and Sentiment Signals

12 April, 2025

Team Members

Ahmed Ibrahim

Leader

Ahmed Awelkair

Member

Amjad Alzain

Member

Alyssa Atmasava

member

Overview

- Quantitative Trading ("Quant"): Using mathematical models, data analysis, and automated algorithms to make trading decisions.
- The Crypto Arena: Offers unique opportunities & challenges:
  - High Volatility & Rapid Evolution
  - 24/7 Market Operation
  - Novel Data Sources (On-Chain, Social Sentiment)

Overview

- The Goal - Generating "Alpha": Finding consistent, predictable trading edges that outperform the general market movements. It's about skill, not just luck or riding a bull run.
- The Challenge: Alpha is elusive. Markets are complex, noisy, and competitive. Success requires sophisticated, data-driven strategies.
- Our Focus: This presentation outlines our approach to systematically tackling this challenge using advanced Machine Learning techniques.

Introduction

The Alpha Generation Challenge

- **Problem Statement:** Develop a robust Machine Learning model analyzing diverse on-chain & market data (≤ 1-day interval) to generate a high-performance alpha trading strategy (SR ≥ 1.8, MDD ≥ -40%, Freq ≥ 3%).
- **Core Difficulty:** Extracting persistent predictive signals ("alpha") from notoriously noisy and non-stationary crypto market data, requiring methods that identify implicit indicators and deterministic patterns.
- **Requirement:** The model must effectively leverage techniques like HMMs for pattern recognition and adapt to market complexities.

Proposed Solution

- **Hypothesis:** We hypothesize that effective alpha generation requires understanding market structure, identifying the prevailing regime, and gauging collective sentiment concurrently. Relying on isolated factors often yields incomplete insights and limits predictive power.
- **Innovation Pillars:** Our framework integrates three distinct but complementary analytical components:
  - **Deep Structural Feature Learning (LSTM AE):** Capturing sequential structure.
  - **Market Regime Identification (HMM):** Providing market context.
  - **Quantitative Sentiment Analysis (NLP):** Gauging market psychology.
- **Value Proposition:** This synergistic integration offers a more robust and adaptive approach than siloed single-method approaches, directly addressing the need for sophisticated pattern recognition and implicit indicator extraction.

Proposed Solution

Contextual Adaptation & Sentiment Integration

- **Adaptive Strategy via HMM Regimes:**
  - Fulfills the recommendation for HMMs by identifying distinct market states (e.g., Bull Trend, Bear Trend, Consolidation) based on price dynamics (Volatility, Momentum).
  - **Key Innovation:** Enables **context-aware decision making**; strategy logic dynamically adapts based on the prevailing regime, enhancing robustness across different market conditions.
- **Behavioral Insights via NLP Sentiment:**
  - Leverages NLP on textual data (Tweets/News) to quantify market sentiment.
  - **Key Innovation:** Incorporates a **quantitative behavioral dimension**, generating features like Sentiment Momentum that capture crowd psychology shifts potentially leading price action.

Proposed Solution

Deep Learning for Implicit Indicators & Synergy

- **Implicit Indicator Extraction via LSTM Autoencoder:**
  - Applies unsupervised deep learning to capture complex, non-linear sequential dependencies within core numerical data (on-chain flows, market data).
  - **Key Innovation:** Automatically learns latent structural features – powerful "implicit indicators" reflecting underlying dynamics that are difficult to manually engineer.
- **Framework Synergy & Originality:**
  - The core novelty lies in the **purposeful integration** of these diverse techniques. AE (Structure) + HMM (Regime) + NLP (Sentiment) provide **complementary perspectives.**
  - The final ML model synthesizes these inputs, creating a system more adaptive and potentially more predictive than the sum of its parts. This architecture directly tackles the complexity and noise inherent in the target data.

Technical Details: Data Pipeline & Feature Synthesis

- Required Data Sources:
  - On-Chain Metrics (e.g., flow_mean, transactions_count_flow from CryptoQuant/Glassnode)
  - Market Data (OHLCV - Essential Addition)
  - Textual Data (Twitter, News Feeds - Essential Addition)
- Processing: Rigorous timestamp alignment (≤ 1-day interval, e.g., 4H), NaN handling, sequence generation (for AE), feature scaling (fit on train).
- Feature Synthesis: Combine outputs into a unified feature set per timestamp:
  - AE Latent Features (Vector)
  - HMM Regime State (Categorical)
  - NLP Sentiment Score(s) (Numerical)

	A	B	C	D	E
1	start_time	datetime	flow_mean	flow_total	transactions_count_flow
2	1.70407E+12	1/1/2024 0:00	0.02986952	0.17921713	6
3	1.70407E+12	1/1/2024 1:00	0.00267839	0.01607036	6
4	1.70407E+12	1/1/2024 2:00	1.42911409	10.00379861	7
5	1.70408E+12	1/1/2024 3:00	0.02899176	0.23193404	8
6	1.70408E+12	1/1/2024 4:00	0.01417226	0.05668903	4
7	1.70409E+12	1/1/2024 5:00	0.03984344	0.23906064	6
8	1.70409E+12	1/1/2024 6:00	0.29878131	2.39025049	8
9	1.70409E+12	1/1/2024 7:00	0.20725916	1.24355499	6
10	1.7041E+12	1/1/2024 8:00	0.02327548	0.30258122	13
11	1.7041E+12	1/1/2024 9:00	0.04585581	1.97179964	43
12	1.7041E+12	1/1/2024 10:00	0.09300796	0.55804775	6
13	1.70411E+12	1/1/2024 11:00	0.01106718	0.30988105	28
14	1.70411E+12	1/1/2024 12:00	0.0130725	0.66669772	51
15	1.70411E+12	1/1/2024 13:00	0.0230991	0.73917114	32
16	1.70412E+12	1/1/2024 14:00	0.02700472	1.91733547	71
17	1.70412E+12	1/1/2024 15:00	0.03779271	0.22675628	6

fig. cybotrade_datasource

Technical Details: Model Architectures & Training

- Component Models:
  - **LSTM Autoencoder** e.g with TensorFlow/Keras
  - **HMM**: hmmlearn.hmm.GaussianHMM, [N] states determined via BIC/AIC & validation. Trained on regime indicators.
  - **NLP** e.g., Transformers library with FinBERT
  - Final Predictor: **XGBoost Classifier**
- Training Methodology:
  - AE & HMM trained on initial training portion.
  - Final Predictor: Trained using Walk-Forward Validation on the synthesized feature set against the price-derived target variable.

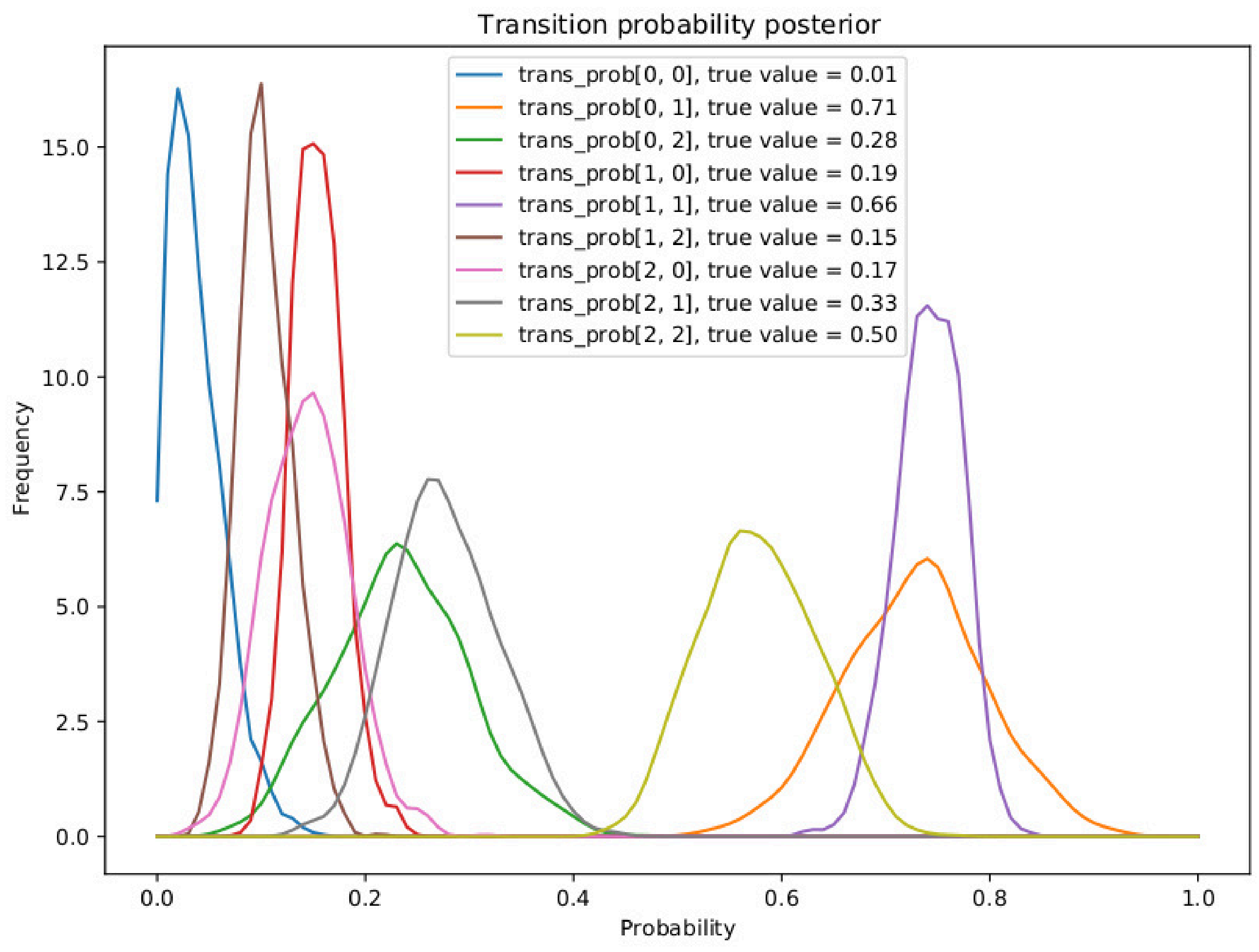

fig. HMM Probabilities

Architecture: Strategy Execution & Backtesting

- Signal Generation: Translate final predictor's output probabilities into preliminary trade signals (e.g., P(Up) > 0.6 -> Prelim Long).
- Execution Logic (HMM Integration): Modulate signals based on HMM state:
  - Example: Permit Long only in Bull/Accumulation states.
  - Example: Reduce size or skip trades in High-Vol/Uncertain states.
  - (This directly addresses the problem statement requirement)
- Backtesting:
  - Engine: [e.g., VectorBT, Backtrader]
  - Fees: Account for 0.06% per trade (0.12% round trip).
  - Periods: Several years backtest, at least one year forward test.
  - Frequency Target: Logic tuned to meet ≥ 3% trade frequency.

1 Predictor Output

2 Probability Threshold

3 HMM State Check

4 Signal (Long/Short/Flat)

5 Apply Fees

6 Record Trade

Future Work & Conclusion

Conclusion: We propose a novel, multi-factor framework combining deep learning, probabilistic modeling, and NLP to tackle the crypto alpha challenge. By integrating structural, regime, and sentiment analysis, this approach aims for robust, adaptive, and high-performing trading signals, contingent on successful integration of required data sources.

- Future Enhancements:
    - Incorporate additional data (options, DeFi).
    - Explore more advanced models (Transformers, Attention, Reinforcement Learning).
    - Develop dynamic risk sizing based on HMM confidence or predicted volatility.
    - Extend to multi-asset portfolio optimization

Thank you