This post outlines LevPredict, a model that represents a collateralized approach to facilitating a parimutuel betting market for binary (and multi category) real-world event outcomes. It focuses on financial engineering aspects of market structure and smart contract UX design.

This is not an ICO or project seeking funding it is just an outline anyone can follow. The model does not require any special token, but it does posit a Smart Contract that creates its own ERC20 tokens to serve as vehicles for the outcome values. See the latest version of the paper in PDF here: http://leverj.io/levpredict.pdf

LevPredict requires a trusted event outcome determination, so a trustless resolution mechanism is beyond the scope of this paper. As a result, this is not at truly trustless prediction market model because the trigger to payout to winners is centrally determined.

To our knowledge this style of parimutuel betting facility model has not been applied to Ethereum (or any other crypto blockchain) for decentralised prediction market tokens. Technically the LevPredict model would apply to any Smart Contract capable blockchains not just Ethereum.

Background

The most efficient facilitation of price discovery in prediction markets is to allow free trading of a raw asset representing the probability of a certain event. You can wrap up risk exposure into all kinds of exotic products: options, futures, insurance products, etc., but these are all just middlemen repackaging the price of risk.

If you use X ETH at price δ to buy Y=X/δ tokens to take a position on an outcome of interest, you should have the flexibility to sell or buy more of the exposure just like any other asset, fully collateralized. As the price oscillates up to the event resolution date, the market absorbs the information in the world to reveal determinants of the outcomes. Just like stock shares and ICO assets are traded openly on the market, so too is there a market price δ reflecting the some probability of the outcome of an event. As long as there are sellers available to pay the buyers (and vice versa), then a market is made.

This model uses a Smart Contract implementation that uses ETH as the collateral and creates ERC20 token assets that take on a market value representing the effective probability between outcomes of a binary event. This paper focuses on binary events, but the model can trivially be expanded to multi-category events (and this is suggested to handle tail event outcomes).

In contrast to a futures model approach, there is no margining or risk management required as the mechanism is fully collateralized. And unlike a bookie, the Smart Contract won't go bankrupt, because it holds the ETH put into the system and pays directly to one outcome.

Throughout this post I'll be illustrating by example using, the event of the 2016 United States election had two outcomes: Trump or Not Trump (effectively Hillary). These outcomes are represented by two ERC20 tokens TUP and NUP.

Figure 1:

How event is birthed and matures

Smart Contract Specs

Primarily you have to specify the resolution date, the number of outcomes (tokens), the ETH price of the token pairs (or groups for n>2), and the mechanism to resolve which triggers the payout to winning outcome (this is a big decision, the trigger is in the contract, but someone/something has to pull it). Decimals should be 1 to avoid dust payouts.

People send ETH to the Contract, receive ERC20 tokens representing different event outcomes, and if people send the tokens back (in proportion) to the Contract, it sends ETH back. Then at resolution date an outcome is chosen as winner. The token holders of the winning outcome then get sent back to the Contract, which disburses the ETH pro rata.

Let's say you create: LevPredict Trump-2016 Event Contract which goes through three periods:

- t0

=contract creation, months ahead ideally to get market in price discovery mode early

- t1

=outcome category tokens generating and being redeemed and trading on secondary market in period (TUP and NUP)

- t2

=resolution date (in blockheight), e.g, January 20. 2017 (Inauguration)

Supply creation mechanism

When you send ETH to the contract, the contract issues two ERC20 tokens back to you:

- TUP: Trump United States President 16
- NUP: No Trump United States President 16

So given rate r (in ETH) for token creation, the Contract has received at any given time a net total of Y ETH:

TUP supply Q_TUP=Y/r

NUP supply Q_NUP=Y/r

TUP and NUP supply are always equal. In our example we take r=1.

Figure 2

: Separation of assets in Smart Contract from the secondary market for TUP & NUP

Convertibility window for arbitrage

Since 1 ETH in results in 1 TUP and 1 NUP, the Contract also will return 1 ETH in exchange for 1 TUP and 1 NUP sent together. This forces the market to trade rationally since people could arbitrage any deviation from the sum of the market value of the two tokens by sending ETH to create new tokens and sell down the market.

Figure 3

: Smart contract as a convertibility window at fixed rate TUP and NUP for ETH.

Low prices

Take the case of TUP trading at 0.40 ETH and NUP trading at 0.50 ETH.

One could buy TUP and NUP together for 0.90 and redeem from the smart contract at face value to earn 1.0 ETH.

High prices

Take the case now of TUP trading at 0.55 ETH and NUP trading at 0.54 ETH.

One could send 1 ETH to the Contract, create 1 TUP and 1 NUP, and sell for 1.09, earning 0.09 ETH arbitrage per token.

This keeps the system fundamentally in balance with respect to ETH flows and token pricing on the market.

Pre-Settlement Market Expectations

Since NUP and TUP are ERC20 tokens, they can be traded anywhere: on DEXes or centralised markets that accept Ethereum assets. They rationally should trade between: (0, r] ETH since nobody would buy TUP or NUP for more than the price r that you can get from the Contract.

At price r=1, the market value takes on a reflection of the decimal probability of the outcome occurring. With enough economic/insurable risk/speculation interest associated with the event then it will attract liquidity.

Market participants who did not generate their own tokens may enter the secondary market and speculate and participate in the token market as a prediction market. One can continuously receive market signals of the token value indicating the outcome risk of occurring.

As the world moves closer in time to the event from the present time, the value of the tokens will gravitate toward the true probability at any given time of the event having a particular outcome. The smart contract will programmatically disburse to the winning outcome so there is cryptographic certainty (to the extent the resolution mechanism is trustless) that no matter what price you pay to obtain TUP or NUP, you will receive the face value of 1 ETH if the outcome occurs. This solidifies the credibility of the value of the tokens and reduces uncertainty on the valutaion.

Settlement

Upon resolution date t2:

:

- Smart contract will pay out (or be redeemable for) all ETH to TUP holders if Trump is United States president on Jan 20, 2017, nothing if not.
- Smart contract will pay out (or be redeemable for) all ETH to NUP will if Trump is NOT United States president on Jan 20, 2017, nothing if he is.

In general when determining outcomes, assume only one or the other can happen (i.e., mutually exclusive), and that there is no ambiguity in event outcome. Design of the event is outside of the scope of this paper, but for simplicity, could focus only

on events that are very popular among the global bookmakers such that it is relatively easy to determine consensus of outcome.

This creates a market where people use ETH and ERC20 assets to make unlevered bets on binary event outcomes. The contract can be coded open ended such that a group of trusted parties gather to verify the event outcome after it has occurred.

Once the resolution is verified in the Contract, holders of the winning event get full payout and NUP gets nothing. Any further ETH sent to the Contract is rejected.

Upon settlement block being reached, Contract has Y_end ETH that is distributed pro rata to all TUP holders at price of TUP/ETH = 1 (either via redemption or snapshot airdrop).

Payout can be done in many ways but two main approaches would be suggested:

1. The winning token is chosen in the Smart Contract, and users must send their tokens back to the Smart

Contract to redeem their ETH.

1. A “snapshot” is taken at the resolution date, at which time the Smart Contract disburses the ETH to all account holders at the specific block.

The first model is preferable because it is more practical in making the user pay for the gas, and then the LevPredict contract is more of a utility people choose to interact with to generate tokens and disburse.

Throughout the period, many ETH may have gone in (when people create TUP and NUP) and out (when people destroy TUP and NUP), but Y_end remains at t2 as no further generation would be allowed. In the second model, Y_end goes out of the contract to distribute to TUP. Gas could be user paid or pre-funded through a pool “rake”.

User journey for betting on outcome

Inception Route

The way a user would make a simple bet on Trump winning would be to send ETH to the Contract, receive 1 TUP and 1 NUP, and sell their NUP tokens to other people who believe Hillary will win. Those who believe Hillary will win will provide the Supply sell pressure on TUP market.

For example say TUP is trading at P_TUP= 0.40 and P_NUP is trading at 0.60 on different exchanges like Leverj, IDEX, Binance, and EtherDelta.

User starts with 1 ETH.

User sends 1 ETH to get 1 TUP and 1 NUP. You sell your 1 NUP for 0.60 ETH, and you use your 0.60 ETH to buy 1.5 TUP (0.6/P_TUP).

After election day the “good news” comes that Trump won. Even though the resolution date is months away,

If the user chooses to hold until settlement, then the contract pays out at the resolution date block where a “snapshot” of sorts is taken to see what addresses are holding TUP and NUP in a given block.

The contract then sends 1.5 ETH in proportion to the 1.5 TUP tokens held in the user’s Ethereum address.

User makes 0.5 ETH on the bet.

Purely Secondary Market Trader

Another user maybe has a lot of ETH and just wants to make a bet on Trump winning. He would not interact with the smart contract, but would just go to an exchange that trades ERC20 tokens and buy what he considers to be a fair price for a return in case Trump wins.

Say market is trading at the above prices, P_TUP=0.40.

User sends 1 ETH to bet on Trump at this price. It buys 2.5 TUP

(which pay out 2.5 ETH if Trump wins).

An effort would need to be made to avoid people scamming so that the TUP and NUP tokens can’t be sent around and duping people into thinking there is value even after the settlement payout has occurred.

Decentralised exchange trading of LevPredict

It bears repeating that LevPredict tokens are fully collateralized. The model poses no system risk to exchanges that list, for example, TUP/ETH and NUP/ETH pairs. They would be free to list them for trading and just earn execution fees in ETH with

no exposure to the underlying volatility.

Additionally, because they are ERC20 tokens they could be trustlessly handled in your favourite trustless exchange or protocol. It is truly decentralised in that anybody can create the tokens if they send ETH to the Smart Contract and trade it wherever they want.

Beyond Binary: Multiple Categories

This is a bit trickier in terms of event design, if there is surely only 3 types of outcomes it can be a functioning market, but if there is ambiguity it can be hard to facilitate a liquid market.

However, in financial engineering terms, one can easily create n > 2 categories with the same convertibility window and settlement to single outcome. Multiple tokens would make it a bit more difficult to arbitrage. The smart contract would require all three, or all n tokens to be submitted to redeem equal amount in ETH (in time t1).

Issues

Liquidity issues

Although the Contract convertibility window does provide arbitrage for summed values deviating, there is no easy way for marketmakers to manage exposure on prediction market outcomes. In my years working with crypto exchanges the number one issue after proper tech is having liquidity. For financial assets with a liquid spot market, market makers can create short and long position adjustments in response to trading activity.

This exposure is more difficult in prediction markets. As a result, liquidity would be an ongoing issue. There is , however, for many events, a global bookmaking market where casinos are offering odds on events, which one could view as a financial asset with a payoff that can adjust exposure.

If this is not obvious at first, think about it in purely mathematical terms: say you want to bet at a bookie 1 ETH that Trump wins, he offers you odds payout of 2.4x. You would earn 2.4 ETH from it if Trump wins and end up with 3.4 ETH.

This is identical to buying 3.4 TUP at TUP/ETH = 0.30 for 1 ETH. This is the market value of gaining exposure to the possibility of this event.

If the event has widespread appeal, then quants can provide liquidity and exploit differentials between different bookmaking markets around the world. LevPredict tokens would be the freest and most secure market for an event because you are guaranteed to have bookie solvency in the Smart Contract.

Event Definition

When defining a particular event, care must be taken to ensure mutual exclusivity. For most practical applications (politics, sports events with clear rules) there is reasonable certainty and minimal ambiguity regarding resolution.

Anomalous / Unaccounted Outcomes

In practice events can be shocking and not result in any of the predicted outcomes. There is no obvious way to refund token holders because they would be traded around and switching hands so much where people took different exposure. Thus, it is not clear how it would disburse fairly. A 50/50 split would benefit speculators of the cheaper outcome on the market. Potentially could disburse to all those who have overlap of TUP and NUP for ETH, and then split the rest to the people.

Cheekier way would be to create a third token, lets call it 00P, which would be issued in addition to NUP and TUP,. and be paid out fully in anomalous unaccounted outcomes. We treat it as “00” in Roulette.