CS164 Final Project: Extended Portfolio Optimization

Marina Levay

December 19, 2025

Contents

1 Introduction

This project extends the classical Markowitz mean–variance portfolio optimization framework introduced in Session 20 – Quadratic Programming [8, 1] to event-based financial markets, such as those traded on platforms like Kalshi [2]. In these markets, traders buy binary event contracts that pay a fixed amount if a specific outcome occurs, and profit when the market price is below their own estimated probability.

Rather than allocating capital across many assets at a single point in time, we model a single event contract as an asset whose value evolves. The decision variable is the size of the position held in the same contract across multiple periods as market prices and beliefs change.

Expected returns are defined as the difference between a subjective forecast probability and the market-implied probability, while risk is captured through a variance term reflecting unresolved event uncertainty. The resulting optimization problem balances expected edge against risk and transaction costs under realistic constraints, including finite capital, long-only positions (no short selling, so position sizes are constrained to be nonnegative), and periodic re-optimization. The model remains a convex optimization problem with linear constraints and is solved using CVXPY.

2 Solution Specification

2.1 Decision variables

Let $u_i \geq 0$ be the dollar position held in a single event contract at time $i = 1, \dots, n$. The variable u_i represents the trader's exposure to the event at time i. We denote the initial position by $u_0 = w$ and impose what is known as terminal liquidation $u_{n+1} = 0$: at the end of the modeled decision horizon, any remaining position must be closed, and its cost should be accounted for. In practice, this means that contracts resolve at a fixed time and cannot be held indefinitely.

2.2 Data and parameters

At each time i, we use the following quantities:

- $p_i \in [0, 1]$: the observed Kalshi YES outcome price at time i, interpreted as the market's implied probability that the event occurs,
- $q_i = Q_0 \in [0, 1]$: a constant subjective probability chosen by the trader and held fixed over the time window,
- $y_i = q_i - p_i$: the expected edge at time i, measuring how much the trader's belief differs from the market price,
- $\sigma_i^2 = p_i(1 - p_i)$: a simple variance proxy based on the Bernoulli payoff of the event contract, which reflects how uncertain the outcome still is,
- $\lambda > 0$: a risk-aversion parameter that limits how large a position the model is willing to take,
- $c > 0$: a transaction cost parameter that penalizes changing the position too aggressively over time.

2.3 Optimization problem

The trader seeks to choose a sequence of position sizes that maximizes expected cumulative profit while accounting for both risk and transaction costs. This leads to the following optimization problem:

$$\varphi(w) = \max_{u_1, \dots, u_n} \quad \sum_{i=1}^{n} \left(y_i u_i - \lambda \sigma_i^2 u_i^2 - c\,|u_i - u_{i-1}| \right) - c\,|u_{n+1} - u_n| \tag{1}$$

$$\text{s.t.} \quad u_i \geq 0, \quad i = 1, \dots, n, \tag{2}$$

where $u_0 = w$ and $u_{n+1} = 0$.

As we will show in the questions below, the objective function is concave in the decision variables $(u_1, \dots, u_n)$. The expected return term is linear, the risk term is a negative quadratic with $\sigma_i^2 \geq 0$, and the turnover penalty is the negative of a convex absolute-value function ($|u_i - u_{i-1}|$ is convex in $u_i - u_{i-1}$ and penalizes the magnitude of position changes). The feasible set defined by the nonnegativity constraints is a closed, convex polyhedron. Therefore, the problem is a convex optimization problem and admits a global optimum, consistent with the general theory of convex optimization discussed in Session 13 [6].

2.4 Nature of the optimization problem and relation to course topics

From the perspective of quadratic programming, the core of the objective function consists of quadratic and linear terms. If the turnover penalties $|u_i - u_{i-1}|$ were absent, the problem would reduce to a standard quadratic program of the type studied in Session 20 [8] and in the lecture notes by Heider [1], with a quadratic objective and linear inequality constraints. In that case, the problem could be analyzed entirely using quadratic programming techniques and KKT conditions.

The inclusion of absolute-value turnover penalties means that the full problem is no longer a pure quadratic program, since the objective is not everywhere quadratic. However, the problem remains convex, and its structure is closely related to quadratic programming in two important ways. First, the absolute-value terms are convex and can be handled by introducing auxiliary variables, allowing the problem to be reformulated as a quadratic program with additional linear constraints. Second, for fixed values of the dual variables introduced to handle these terms, the resulting subproblems are concave quadratic maximization problems that can be solved using the same quadratic programming techniques discussed in [8, 1].

The problem also admits a dual representation, which connects directly to the material on duality covered in Session 17 [7]. In this formulation, dual variables arise from the turnover penalties and can be interpreted as shadow prices associated with changes in position size over time. This dual representation provides insight into the sensitivity of the optimal value with respect to the initial position, and clarifies the financial role of transaction costs, and this is consistent with the interpretation of dual variables discussed in [7] and illustrated geometrically in [5].

Overall, this formulation shows how the different parts of the course fit together: quadratic programming techniques are used to solve the quadratic components of the problem, while convex duality provides the framework for handling nonsmooth convex penalties and for interpreting sensitivity and shadow prices.

3 Analysis and Theoretical Questions

3.1 Question 1: Dual formulation

Question. Find a dual representation for the value function $\varphi(w)$.

Solution. To derive a dual representation of $\varphi(w)$, we view the problem through the duality framework introduced in Session 17 [7]. The objective defines a concave maximization problem (as shown in Question 2) with affine inequality constraints $u_i \geq 0$. By negating the objective, the problem can be written in the standard form of a convex minimization problem.

Since there exists a strictly feasible point for the inequality constraints (for example, choosing any $u_i > 0$), Slater's condition holds. This guarantees strong duality, meaning that the primal and dual problems attain the same optimal value and that it is valid to analyze the problem through its dual representation.

The main difficulty in forming the dual comes from the turnover penalties $|u_i - u_{i-1}|$, which are convex but not differentiable. Following the duality framework, these terms can be rewritten using their support-function representation (where $d := |u_i - u_{i-1}|$):

$$c|d| = \max_{|z|\le c} z\, d \quad \Longrightarrow \quad -c|d| = \min_{|z|\le c}(-z\, d)$$

This representation follows from the geometry of convex functions discussed in Session 13 [6]. Applying this identity to each increment $u_i - u_{i-1}$, including the terminal increment $u_{n+1} - u_n$, introduces dual variables $z_1, \ldots, z_{n+1}$ with bounds $|z_i| \le c$. Substituting into the objective gives us this saddle-point formulation:

$$\varphi(w) = \min_{\substack{|z_i|\le c\\ i=1,\ldots,n+1}} \ \max_{u_1,\ldots,u_n\ge 0} \left\{ \sum_{i=1}^{n} \left(y_i u_i - \lambda \sigma_i^2 u_i^2\right) - \sum_{i=1}^{n+1} z_i(u_i - u_{i-1}) \right\}$$

For fixed z, the inner maximization separates across time. Each u_i appears in a concave quadratic expression of the form

$$(y_i - z_i + z_{i+1})u_i - \lambda\sigma_i^2 u_i^2,$$

together with the linear term $z_1 w$. These one-dimensional subproblems can be solved analytically using quadratic programming techniques discussed in Session 20 [8] and in Heider [1]. Solving each subproblem gives us:

$$\max_{u_i\ge 0} = \frac{(y_i - z_i + z_{i+1})^2}{4\lambda\sigma_i^2}, \quad (t) := \max(t, 0)$$

Substituting these expressions back into the objective gives the dual representation

$$\varphi(w) = \min_{\substack{|z_i|\le c\\ i=1,\ldots,n+1}} \left[z_1 w + \sum_{i=1}^{n} \frac{(y_i - z_i + z_{i+1})_+^2}{4\lambda\sigma_i^2} \right]$$

Because Slater's condition holds, this dual formulation attains the same optimal value as the primal problem.

3.2 Question 2: Concavity and subgradients

Question. Show that $\varphi(w)$ is concave in w, and find a subgradient of $-\varphi$ at w. If φ is differentiable at w, determine its gradient.

Solution. From the dual formulation, for any fixed choice of dual variables z, the value function can be written as

$$\varphi(w) = z_1 w + g(z),$$

where $g(z)$ does not depend on w. This expression is affine in w. As explained in Session 17 [7], when a value function is expressed as the minimum over a family of affine functions, the resulting function is concave.

Therefore, $\varphi(w)$ is concave in w. Geometrically, $\varphi(w)$ corresponds to the lower envelope of a collection of straight lines, which is a concave function, as illustrated in Kochenderfer and Wheeler [5].

To compute a subgradient, let $z^\star$ be a dual minimizer at w. Since w appears only through the term $z_1 w$, the slope of the active affine function is $z_1^\star$. As a result,

$$z_1^\star \in \partial\varphi(w), \qquad -z_1^\star \in \partial(-\varphi)(w).$$

If the dual minimizer is unique, then φ is differentiable at w, and the gradient is given by

$$\nabla\varphi(w) = z_1^\star, \qquad \nabla(-\varphi)(w) = -z_1^\star.$$

3.3 Real-World Data Interpretation

We implemented the optimization problem in CVXPY and tested it using live, unauthenticated market data from Kalshi [3] on the open market **"Will an LLM beat a grandmaster in chess this year?"** [4].

For this contract, we interpret the market YES outcome price (quoted in cents) as an implied probability $p_i \in [0, 1]$ and compute a variance proxy $\sigma_i^2 = p_i(1 - p_i)$. This aligns with the Bernoulli uncertainty associated with an unresolved binary payoff.

Since the true model forecast q_i is application-dependent (and requires a separate predictive model that converts information about LLM and chess performance into a probability), we use the simplest baseline forecast: a constant subjective belief $q_i = Q_0$ for all i. Under this choice, the edge is $y_i = Q_0 - p_i$.

This constant-belief model gives us a clear practical decision rule. If $Q_0 > p_i$ (the trader believes the contract is underpriced), then $y_i > 0$ and the optimizer increases exposure; if $Q_0 \leq p_i$, then the long-only constraint implies the optimal exposure is reduced toward zero. The risk aversion term $\lambda\sigma_i^2 u_i^2$ prevents the strategy from taking large positions even if $Q_0 > p_i$, while the turnover penalty $c|u_i - u_{i-1}|$ discourages frequent rebalancing.

In one representative run on `KXLLMCHESS-26` with the constant-belief forecast (with $Q_0 = 0.05$ chosen above the observed market YES price level, which was approximately \$0.02 during the polling window), the solver returned the following numerical results:

$$\begin{aligned} \text{primal status} &= \texttt{optimal}, & \varphi_{\text{primal}}(w) &= 0.017551020408163264, \\ \text{dual status} &= \texttt{optimal}, & \varphi_{\text{dual}}(w) &= 0.017551020408163268 \end{aligned}$$

The corresponding optimal primal positions were approximately constant over time,

$$u_i^* \approx 0.15306122, \quad i = 1, \dots, n,$$

and the first entries of the dual optimizer satisfied $z_i^* \approx -0.01$ (which is consistent with the bound $|z_i| \leq c$ when $c = 0.01$). The primal–dual gap in this run was on the order of 10^{-18}, consistent with strong duality.

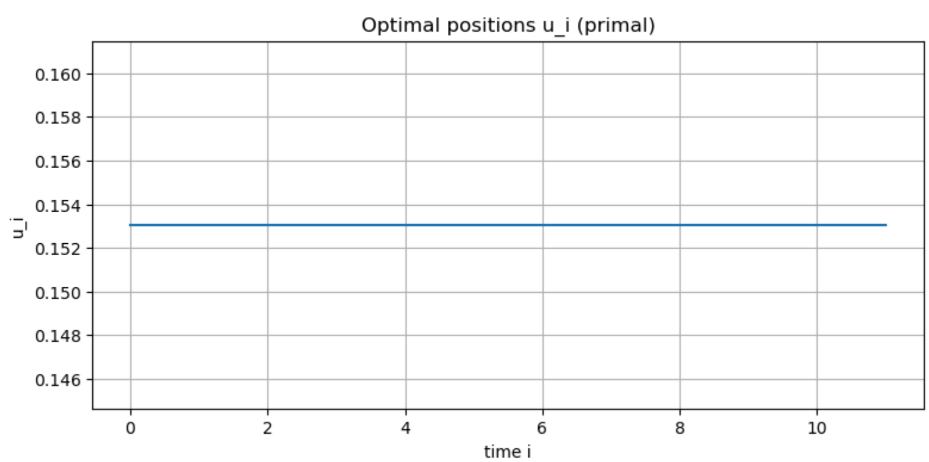

Figure 1: Optimal primal positions $u_i^\star$ across time steps (approximately constant over the polling window).

Practically, $u_i^* \approx 0.153$ should be interpreted in the same units as the initial position w. In our notebook implementation, we use a normalized initial position $w = 1$, so $u_i^* \approx 0.153$ corresponds to maintaining about 15.3% of the initial position as exposure throughout the sampled period. The near-constant shape reflects that the inputs p_i (and thus y_i and σ_i^2) were nearly constant during the polling window, so the optimizer has no incentive to frequently rebalance. The dual values in the constraint (here $z_i^* \approx -c$) indicate that transaction costs are active at the optimum. In particular, z_1^* is the sensitivity (subgradient) of $\varphi(w)$ with respect to the initial position w.

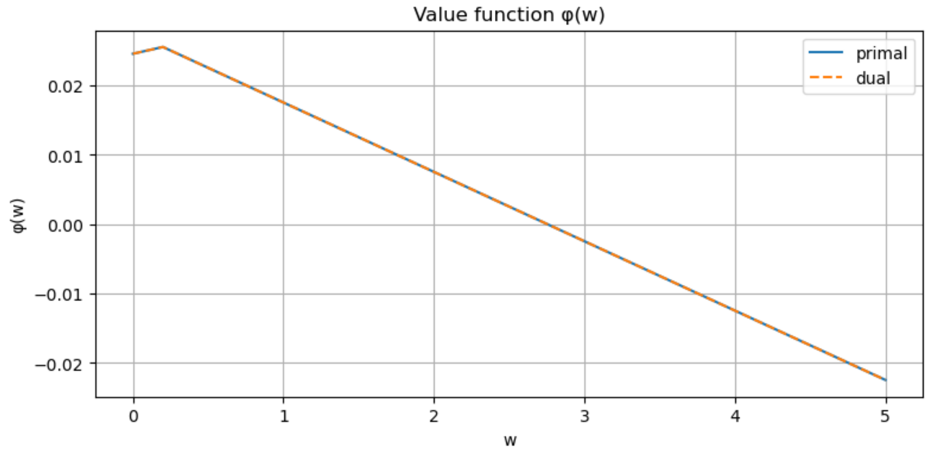

Figure 2: Value function $\varphi(w)$ estimated from Kalshi polling data: primal and dual values overlap up to numerical tolerance, confirming strong duality.

Figure 2 plots the value function $\varphi(w)$ over a grid of initial positions w by resolving both the primal and dual problems at each w. The near-perfect overlap between the primal and dual curves also confirms strong duality for this case. The plot is concave (as demonstrated in Question 2) and, in this short polling window, nearly linear: the slope is approximately constant because the optimal dual variable z_1^* (a subgradient of φ) remains close to its constraint bound. The sign change of $\varphi(w)$ across the grid is meaningful: values above zero correspond to parameter regimes where the model finds a net positive risk-adjusted opportunity under the chosen Q_0, while values below zero indicate that the risk and transaction penalties dominate.

4 Conclusion

This project extends mean–variance optimization to dynamic trading in a single binary event contract with long-only positions, risk control through the Bernoulli variance, and turnover costs. Using a support-function representation of the absolute-value penalties, we derive a dual formulation in which $\varphi(w)$ is the minimum of affine functions of w, implying concavity and giving us a sensitivity interpretation through z_1^*.

On live Kalshi data for `KXLLMCHESS-26`, the primal and dual values matched to numerical tolerance (strong duality). With normalized initial capital $w = 1$ and $Q_0 = 0.05$ (versus market YES price near 0.02), the model recommends a stable long exposure $u_i^* \approx 0.153$ (about 15.3% of capital), with objective value $\varphi(w) \approx 0.0176$.

The dual sensitivity $z_1^* \approx -0.01 \approx -c$ indicates transaction costs are binding, so rebalancing is only justified when the incremental gain from a larger edge y_i outweighs the turnover penalty $c\,|\Delta u|$ (and the additional risk penalty). In practice, this means small fluctuations in prices or beliefs should not trigger trades: the edge must move enough to compensate for the cost of changing the position.

5 Appendix

References

[1] Heider, J. (2015). *Quadratic programming*. Lecture notes.

[2] Kalshi. (n.d.). *Kalshi prediction markets*. Retrieved from `https://kalshi.com/`

[3] Kalshi. (n.d.). *Kalshi API documentation*. Retrieved from `https://docs.kalshi.com/welcome`

[4] Kalshi. (n.d.). *LLM chess market*. Retrieved from `https://kalshi.com/markets/kxllmchess/llm-chess/kxllmchess-26`

[5] Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for optimization*. MIT Press.

[6] Minerva University. (n.d.). *CS164 Session 13: Convex sets and functions*. Retrieved from `https://forum.minerva.edu/app/courses/3723/sections/12804/classes/96775`

[7] Minerva University. (n.d.). *CS164 Session 17: Duality*. Retrieved from `https://forum.minerva.edu/app/courses/3723/sections/12804/classes/97500`

[8] Minerva University. (n.d.). *CS164 Session 20: Quadratic programming*. Retrieved from `https://forum.minerva.edu/app/courses/3723/sections/12804/classes/98060`

[9] Slama, T. (2023, July 4). *The art of linear programming* [Video]. YouTube. `https://www.youtube.com/watch?v=E72DWgKP_1Y`

5.1 CVXPY

Being mindful of the length of this report, all code is in a notebook that has been included along with the report in a `.zip` file.

5.2 AI Statement

I used Gemini 3 Pro on my coding IDE to more efficiently fix bugs when working with Kalshi's API documentation. I also used Grammarly AI to avoid typos and grammatical mistakes.