Sparse Mean-Reverting Portfolios via Penalized Likelihood Optimization

by Galvin Adriel

February 16, 2026

Abstract

This short LaTeX paper shows the outline derivation and math for my personal project on the mean reversion of portfolios. Here I also eleborate the moments for using other mean reverting type models of diffusion, in particular interest rates of diffusions, thus generalizing the NLL into the moments of each diffusion [1].

1 Mathematical Derivation summary

We start from the continuous-time Ornstein–Uhlenbeck SDE [2]

$$dx_t = \mu(\theta - x_t)\,dt + \sigma\,dB_t$$

Exact solution is:

$$x_t = x_{t-1}e^{-\mu\Delta t} + \theta\left(1 - e^{-\mu\Delta t}\right) + \varepsilon_t$$

where:

$$\varepsilon_t \sim \mathcal{N}\left(0, \tilde{\sigma}^2\right)$$

and:

$$\tilde{\sigma}^2 = \frac{\sigma^2}{2\mu}\left(1 - e^{-2\mu\Delta t}\right)$$

The likelihood function has a density of:

$$f(x_t \mid x_{t-1}) = \frac{1}{\sqrt{2\pi\tilde{\sigma}^2}}\exp\left(-\frac{\left(x_t - x_{t-1}e^{-\mu\Delta t} - \theta(1 - e^{-\mu\Delta t})\right)^2}{2\tilde{\sigma}^2}\right)$$

Over T periods:

$$\mathcal{L} = \prod_{t=1}^{T} f(x_t \mid x_{t-1})$$

[1] Click Here to see GitHub

[2] See Appendix for OU moments

Take the log and multiply by -1, we get:

$$-\log\mathcal{L} = \frac{T}{2}\log(2\pi) + \frac{T}{2}\log(\tilde{\sigma}^2) + \frac{1}{2\tilde{\sigma}^2}\sum_{t=1}^{T}\left(x_t - x_{t-1}e^{-\mu\Delta t} - \theta(1-e^{-\mu\Delta t})\right)^2$$

Recall that portfolio is via

$$x_t = S_t w$$

where $S_t \in \mathbb{R}^m$ are asset prices and $w \in \mathbb{R}^m$ are portfolio weights. The matrix is defined as:

$$A(\mu) = S_{1:T} - e^{-\mu\Delta t}S_{0:T-1} \equiv x_t - x_{t-1}e^{-\mu\Delta t}$$

Define:

$$y(\theta,\mu) = \theta(1-e^{-\mu\Delta t})\mathbf{1}$$

The square error becomes:

$$\|A(\mu)w - y(\theta,\mu)\|^2$$

Define new parameters:

$$c = e^{-\mu\Delta t}$$

$$a = \tilde{\sigma}^2 = \frac{\sigma^2(1-e^{-2\mu\Delta t})}{2\mu}$$

Use $e^x \approx 1 + x$ as approximation:

$$a \approx \Delta t\sigma^2$$

$$c \approx 1 - \mu\Delta t$$

The average log-likelihood is $\ell = \log\mathcal{L}/T$, such that using the notation beforehand:

$$\ell(a,c,\theta,w) = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(a) - \frac{1}{2Ta}\|A(c)w - y\|^2 \tag{3.3.1}$$

Drop the constant $T/2\log(2\pi)$:

$$\min_{\mu,\sigma^2,\theta}\ \frac{1}{2}\log(\tilde{\sigma}^2) + \frac{\|A(\mu)w - y(\theta,\mu)\|^2}{2T\tilde{\sigma}^2}$$

The final optimization problem is recalling that $y = \theta(1-e^{-\mu\Delta t})\mathbf{1} = \theta(1-c)\mathbf{1}$:

$$\min_{a,c,\theta,w}\ \frac{1}{2}\ln(a) + \frac{\|A(c)w - \theta(1-c)\mathbf{1}\|^2}{2Ta}$$

subject to:

$$\|w\|_1 = 1$$

Promoting sparsity:

$$\|w\|_0 \leq \eta$$

To promote stronger mean reversion, add a factor $+\gamma c$, such that the final penalized problem:

$$\min_{a,c,\theta} \frac{1}{2}\ln(a) + \frac{\|A(c)w - \theta(1-c)\mathbf{1}\|^2}{2Ta} + \gamma c$$

subject to:

$$\|w\|_1 = 1, \quad \|w\|_0 \leq \eta$$

The variable projection is defined as the value function of:

$$f(w,a,c,\theta) = \frac{1}{2}\ln(a) + \gamma c + \frac{\|A(c)w - \theta(1-c)\mathbf{1}\|^2}{2Ta}$$

We can nest the minimizations via the work flow of (in a nutshell) :

$$\arg\min_{\theta} f(\cdot) \to \arg\min_{c,\theta} f(\cdot) \to \arg\min_{a,c,\theta} f(\cdot) \to f_3(w)$$

In which:

$$f_1(w,a,c) = \min_{\theta} f(w,a,c,\theta)$$

$$f_2(w,a) = \min_{c} f_1(w,a,c)$$

$$f_3(w) = \min_{a} f_2(w,a)$$

so that the final reduced problem is :

$$\min_{\|w\|_1=1,\ \|w\|_0\leq\eta} f_3(w)$$

In which we can continue to the gradient descent[3].

2 Gradient Descent

Define:

$$b_1 = BS_{1:T}w, \qquad b_0 = BS_{0:T-1}w$$

and also:

$$u = b_0^\top b_1, \qquad v = \|b_0\|^2$$

2.1 Case 1 : $\gamma = 0$

We obtained:

$$a^* = \frac{1}{T}\|b_1 - cb_0\|^2$$

and:

$$f_3(w) = \frac{1}{2}\left(\ln(a^*) + 1\right) = \frac{1}{2}\left(\ln\left(\frac{\|b_1 - cb_0\|^2}{T}\right) + 1\right)$$

[3] See Appendix for Algebra

The closed form gradient is:

$$\nabla_w f_3(w) = \frac{1}{\|b_1 - cb_0\|^2} \left(BS_{1:T} - BS_{0:T-1}ww^\top - cBS_{0:T-1}\right)^\top (b_1 - cb_0)$$

In compact matrix form is:

$$M(w) = BS_{1:T} - BS_{0:T-1}ww^\top - cBS_{0:T-1}$$

This means:

$$\nabla_w f_3(w) = \frac{1}{\|b_1 - cb_0\|^2} M(w)^\top (b_1 - cb_0)$$

2.2 Case 2 : $\gamma > 0$

We had:

$$a^* = \frac{1}{2T\gamma^2} \left(\|b_0\|^2 - \sqrt{\|b_0\|^4 - 4\gamma^2 \left(\|b_0\|^2 \|b_1\|^2 - (b_0^\top b_1)^2\right)} \right)$$

Define the discriminant as:

$$\Delta = \|b_0\|^4 - 4\gamma^2 \left(\|b_0\|^2 \|b_1\|^2 - (b_0^\top b_1)^2\right)$$

Take the derivative of the square root term:

$$\frac{\partial}{\partial w}\sqrt{\Delta} = \frac{1}{2\sqrt{\Delta}} \frac{\partial \Delta}{\partial w}$$

The derivative of Δ is:

$$\begin{aligned}\frac{\partial \Delta}{\partial w} &= 4\|b_0\|^2 (BS_{0:T-1})^\top b_0 \\ &\quad - 4\gamma^2 \left(2\|b_1\|^2 (BS_{0:T-1})^\top b_0 + 2\|b_0\|^2 (BS_{1:T})^\top b_1 \right. \\ &\quad \left. -4(b_0^\top b_1)\left[(BS_{0:T-1})^\top b_1 + (BS_{1:T})^\top b_0\right]\right)\end{aligned}$$

The derivative of a^* is :

$$\frac{\partial a^*}{\partial w} = \frac{1}{2T\gamma^2} \left(2(BS_{0:T-1})^\top b_0 - \frac{1}{2\sqrt{\Delta}} \frac{\partial \Delta}{\partial w} \right)$$

Chain rule :

$$\nabla_w f_3(w) = \frac{\partial f_3}{\partial a} \frac{\partial a}{\partial w}$$

and:

$$\frac{\partial f_3}{\partial a} = \frac{1}{2a} - \frac{\|b_1\|^2}{2Ta^2} + \frac{(b_0^\top b_1)^2}{2Ta^2 \|b_0\|^2} - \frac{\gamma^2 T}{2\|b_0\|^2}$$

Therefore:

$$\nabla_w f_3(w) = \left(\frac{1}{2a} - \frac{\|b_1\|^2}{2Ta^2} + \frac{(b_0^\top b_1)^2}{2Ta^2 \|b_0\|^2} - \frac{\gamma^2 T}{2\|b_0\|^2} \right) \frac{\partial a^*}{\partial w}$$

2.3 Projection Step

The projection step in closed form is:

$$w^{k+1} = \mathrm{Proj}_{\mathcal{W}}\left(w^k - \delta_k \nabla f_3(w^k)\right)$$

where :

$$\mathcal{W} = \{w : \|w\|_1 = 1,\ \|w\|_0 \leq \eta\}$$

The closed form projection is:

$$\mathrm{Proj}_{\mathcal{W}}(x) = \mathrm{sign}(x) \odot \mathrm{proj}_{\Delta_1^\eta}(|x|)$$

where:

$$\Delta_1^\eta = \{u : u^\top 1 = 1,\ \|u\|_0 \leq \eta\}$$

The algorithm is explained in the table below. We sort $|x|$ descending, keep η largest entries, and find threshold β. The value to find is via ρ, such that:

$$\rho = \max\left\{j : |x|_{(j)} - \frac{1}{j}\left(\sum_{i=1}^{j} |x|_{(i)} - 1\right) > 0\right\}$$

with :

$$\beta = \frac{1}{\rho}\left(\sum_{i=1}^{\rho} |x|_{(i)} - 1\right)$$

and final projection:

$$u_i = \max(|x_i| - \beta, 0)$$

then:

$$w_i = \mathrm{sign}(x_i) u_i$$

The final PGD is:

$$w^{k+1} = \mathrm{sign}(z^k) \odot \mathrm{proj}_{\Delta_1^\eta}\left(|z^k|\right)$$

where:

$$z^k = w^k - \delta_k \nabla f_3(w^k)$$

Algorithm 1 Projected Gradient Descent (PGD) Algorithm

```
    **Input:** $w \in \mathbb{R}^m, S, f_3, \gamma, \eta$
    **Initialization:** $W = \{w : \|w\|_1 = 1, \|w\|_0 = \eta\}$
    **while** not convergent **do**
        $w^k \leftarrow \text{Proj}_W \left(w^{k-1} - \delta_i \nabla_w f_3(w^{k-1}; \gamma, \eta)\right)$
        Recover $a, c, e$ from $w$.
        ($\delta_i$ symbolizing step search).
    **end while**

    **Projection procedure** $\text{Proj}_W(x)$:
      1. Sort $|x|$ so that $|x_1| \geq |x_2| \geq \cdots \geq |x_m|$.
      2. For element $x$ indexed 1 until $\eta$:
         a. Find value $\rho = \max\{j \in [1, \ldots, \eta] : |x_j| - \frac{1}{j}(\sum_{r=1}^{j} |x_r| - 1) > 0\}$.
         b. Define $\beta = \frac{1}{\rho}(\sum_{r=1}^{\rho} |x_r| - 1)$.
         c. **Output:** $u_i$ where $u_i = \max\{|x_i| - \beta, 0\}$.
      3. For vector element $x$ indexed $\eta + 1$ to $m$, $x_{\eta+1:m} = 0$.
```

3 Python

3.1 OU Gradient:

This section breaks down the code and how it connects back to the paper. First The general NLL is :

$$\mathcal{L} = \sum_{t=1}^{T} \left[\frac{1}{2} \log(2\pi) + \frac{1}{2} \log(\sigma_t^2) + \frac{(x_t - \mu_t)^2}{2\sigma_t^2} \right]$$

which can be seen in listing 1. Second, the simplex projection is: The simplex projection via $\Delta_1 = \{w \in \mathbb{R}^m : w^\top 1 = 1, \ w_i \geq 0\}$:

$$\rho = \max \left\{ j : |x|_{(j)} - \frac{1}{j} \left(\sum_{i=1}^{j} |x|_{(i)} - 1 \right) > 0 \right\}$$

$$\beta = \frac{1}{\rho} \left(\sum_{i=1}^{\rho} |x|_{(i)} - 1 \right)$$

$$u_i = \max(|x_i| - \beta, 0)$$

which is reflected in listing 2. The sparsity projection is via: Weight projection of

$$W = \{w : \|w\|_1 = 1, \ \|w\|_0 \leq \eta\}$$

to

$$\text{Proj}_W(x) = \text{sign}(x) \odot \text{proj}_{\Delta_1^\eta}(|x|)$$

. The code is via listing 3. The gradient has form

$$\nabla f_3(w) = \frac{1}{\|b_1 - cb_0\|^2} M(w)^\top (b_1 - cb_0)$$

and

$$\nabla f_3(w) = \frac{\partial f_3}{\partial a} \frac{\partial a}{\partial w}$$

which is listed in listing 4, where $c = \frac{b_0^\top b_1}{\|b_0\|^2}$, $M(w) = BS_{1:T} - BS_{0:T-1} w w^\top - cBS_{0:T-1}$, and $a^* = \frac{1}{2T\gamma^2} \left(\|b_0\|^2 - \sqrt{\Delta} \right)$. Listing 5 shows the gradient descent's main program of :

$$w^{k+1} = \mathrm{Proj}_W \left(w^k - \delta_k \nabla f_3(w^k) \right)$$

Listing 1: Generalized NLL Calculation in Python

```python
def calculate_generalized_nll(observed_data, predicted_mean, predicted_var):
    T = len(observed_data)
    predicted_var = np.maximum(predicted_var, 1e-9)
    term1 = (T / 2.0) * log(2 * pi)
    term2 = 0.5 * np.sum(np.log(predicted_var))
    term3 = np.sum((observed_data - predicted_mean)**2 / (2 * predicted_var))
    return term1 + term2 + term3
```

Listing 2: Simplex Projection Function

```python
def simplex_proj(y):
    n_features = len(y)
    u = np.sort(y)[::-1]
    css = np.cumsum(u)
    ind = np.arange(n_features) + 1
    cond = u - (css - 1.0) / ind > 0
    rho = ind[cond][-1]
    theta = (css[cond][-1] - 1.0) / rho
    return np.maximum(y - theta, 0)
```

showed below:

Listing 3: Projection with L0 Constraint (Eta)

```python
def proj(x, eta):
    w_simplex = simplex_proj(x)
    if np.count_nonzero(w_simplex) > eta:
        threshold = np.sort(w_simplex)[-eta]
        w_simplex[w_simplex < threshold] = 0
        w_simplex /= (np.sum(w_simplex) + 1e-12)
    return w_simplex
```

Listing 4: Gradient of f3 for Optimization

```python
def gradient_f3(w, gamma, T, S_train):
    B = np.identity(T) - np.ones([T,T])/T
    B0 = np.matmul(B, S_train[:T])
    B1 = np.matmul(B, S_train[1:])
    B0_t, B1_t = B0.T, B1.T
    b0, b1 = np.matmul(B0, w), np.matmul(B1, w)
    b01 = np.inner(b0, b1)
    b0_Frob = max(np.inner(b0, b0), 1e-9)

    if gamma == 0:
        c = b01 / b0_Frob
        bcb = b1 - c * b0
        grad = (np.matmul(B1_t, bcb) - c * np.matmul(B0_t,
            bcb)) / max(np.inner(bcb, bcb), 1e-9)
    else:
        b1_Frob = np.inner(b1, b1)
        p = sqrt(max(b0_Frob**2 - 4 * (gamma**2) * (b0_Frob
            * b1_Frob - b01**2), 1e-9))
        q = b0_Frob - p
        grad = (1/max(q, 1e-9)) * np.matmul(B0_t, b0)
    return grad
```

Listing 5: Projected Gradient Descent Implementation

```python
def run_projected_gradient_descent(S_train, w_init, params):
    w = w_init.copy()
    T = S_train.shape[0] - 1
    for i in range(params[’max_iter’]):
        grad = gradient_f3(w, params[’gamma’], T, S_train)
        step = params[’stepsize’] * grad
        w_new = proj(w - step, params[’eta’])
        if norm(w_new - w) < 1e-6:
            break
        w = w_new
    return w
```

Listing 6: Portfolio Negative Log-Likelihood

```python
def portfolio_nll(w, S, delta_t, engine_func, params):
    x = np.matmul(S, w)
    x_current, x_next_actual = x[:-1], x[1:]
    pred_mean, pred_var = engine_func(x_current, delta_t,
        params)
    return calculate_generalized_nll(x_next_actual,
        pred_mean, pred_var)
```

3.2 Other Diffusions:

This subsection is for each diffusions. The table summary is below. Note that the diffusion engine is made in regards to the previous listings of the generalized NLL, so that it doesn't change the moment conditions that effect the optimization of the NLL

Table 1: Discrete Conditional Moments of Short-Rate Models

Model	**Mean** $\mathbb{E}[X_{t+\Delta t} \mid X_t]$	**Variance** $\mathrm{Var}[X_{t+\Delta t} \mid X_t]$
OU	$X_t e^{-\mu\Delta t} + \theta(1 - e^{-\mu\Delta t})$	$\frac{\sigma^2}{2\mu}(1 - e^{-2\mu\Delta t})$
Vasicek	$X_t e^{-a\Delta t} + b(1 - e^{-a\Delta t})$	$\frac{\sigma^2}{2a}(1 - e^{-2a\Delta t})$
CIR	$X_t e^{-a\Delta t} + b(1 - e^{-a\Delta t})$	$X_t \frac{\sigma^2}{a}(e^{-a\Delta t} - e^{-2a\Delta t}) + \frac{b\sigma^2}{2a}(1 - e^{-a\Delta t})^2$
Hull-White	$X_t e^{-a\Delta t} + \frac{\theta}{a}(1 - e^{-a\Delta t})$	$\frac{\sigma^2}{2a}(1 - e^{-2a\Delta t})$
Ho-Lee	$X_t + \text{drift}\,\Delta t$	$\sigma^2 \Delta t$
BDT	$\exp(\ln X_t + \theta\Delta t + \frac{1}{2}\sigma^2\Delta t)$	$(e^{\sigma^2\Delta t} - 1)\cdot$ $\exp(2\ln X_t + 2\theta\Delta t + \sigma^2\Delta t)$
BK	$\exp(e^{-a\Delta t}\ln X_t + \frac{\theta}{a}(1 - e^{-a\Delta t})$ $+\frac{\sigma^2}{4a}(1 - e^{-2a\Delta t}))$	$(e^{v_{\ln}} - 1)\exp(2m_{\ln} + v_{\ln})$

Listing 7: Imports and Header

```python
import numpy as np
""""""" primary diffusion moments for model """""""
```

Listing 8: Ornstein-Uhlenbeck (OU) Engine

```python
def engine_ou(x_current, delta_t, params):
    mu, theta, sigma_sq = params['mu'], params['theta'],
        params['sigma_sq']
    phi = np.exp(-mu * delta_t)
    m_next = x_current * phi + theta * (1 - phi)
    v_next = (sigma_sq / (2 * mu)) * (1 - phi**2)
    return m_next, v_next
```

Listing 9: Vasicek Model Engine

```python
def engine_vasicek(x_t, dt, params):
    a, b, sigma = params['a'], params['b'], params['sigma']
    phi = np.exp(-a * dt)
    m_next = x_t * phi + b * (1 - phi)
    v_next = (sigma**2 / (2 * a)) * (1 - phi**2)
    return m_next, v_next
```

Listing 10: Cox-Ingersoll-Ross (CIR) Engine

```python
def engine_cir(x_t, dt, params):
    a, b, sigma = params['a'], params['b'], params['sigma']
    phi = np.exp(-a * dt)
    m_next = x_t * phi + b * (1 - phi)
    v_next = x_t * (sigma**2 / a) * (phi - phi**2) + (b * sigma**2 / (2 * a)) * (1 - phi)**2
    return m_next, v_next
```

Listing 11: Hull-White Engine

```python
def engine_hw(x_t, dt, params):
    a, theta, sigma = params['a'], params['theta'], params['sigma']
    phi = np.exp(-a * dt)
    m_next = x_t * phi + (theta / a) * (1 - phi)
    v_next = (sigma**2 / (2 * a)) * (1 - phi**2)
    return m_next, v_next
```

Listing 12: Black-Derman-Toy (BDT) Engine

```python
def engine_bdt(x_t, dt, params):
    theta, sigma = params['theta'], params['sigma']
    m_ln = np.log(x_t) + theta * dt
    v_ln = sigma**2 * dt
    m_next = np.exp(m_ln + 0.5 * v_ln)
    v_next = (np.exp(v_ln) - 1) * np.exp(2 * m_ln + v_ln)
    return m_next, v_next
```

Listing 13: Black-Karasinski (BK) Engine

```python
def engine_bk(x_t, dt, params):
    a, theta, sigma = params['a'], params['theta'], params['sigma']
    phi = np.exp(-a * dt)
    m_ln = np.log(x_t) * phi + (theta / a) * (1 - phi)
    v_ln = (sigma**2 / (2 * a)) * (1 - phi**2)
    m_next = np.exp(m_ln + 0.5 * v_ln)
    v_next = (np.exp(v_ln) - 1) * np.exp(2 * m_ln + v_ln)
    return m_next, v_next
```

Listing 14: Ho-Lee Engine

```python
def engine_ho_lee(x_current, delta_t, params):
    drift, sigma_sq = params['drift'], params['sigma_sq']
    m_next = x_current + drift * delta_t
    v_next = sigma_sq * delta_t #linear_growth
    return m_next, v_next
```

4 Appendix : Algebra

4.1 Partial Minimization

The objective is:

$$f(w, a, c, \theta) = \frac{1}{2}\ln(a) + \frac{1}{2Ta}\|A(c)w - \theta(1-c)\mathbf{1}\|^2 + \gamma c$$

Over θ:

$$\frac{\partial f}{\partial \theta} = \frac{1}{2Ta} \cdot 2(1-c)\mathbf{1}^T(\theta(1-c)\mathbf{1} - A(c)w)$$

Set to zero:

$$(1-c)\mathbf{1}^T(\theta(1-c)\mathbf{1} - A(c)w) = 0$$

Expand:

$$\theta(1-c)^2 T = (1-c)\mathbf{1}^T A(c)w$$

Solve:

$$\theta^*(c, w) = \frac{\mathbf{1}^T A(c)w}{T(1-c)}$$

Note that $A(c) = S_{1:T} - cS_{0:T-1}$. Recall that $x_t = S_t w$, so:

$$\begin{aligned} A(c) &= S_{1:T} - cS_{0:T-1} \\ \leftrightarrow A(c)w &= S_{1:T}w - cS_{0:T-1}w \\ \leftrightarrow A(c)w &= x_{1:T} - cx_{0:T-1} \\ \leftrightarrow A(c)w &= x(w)_{1:T} - cx(w)_{0:T-1} \end{aligned}$$

therefore:

$$\theta^*(c, w) = \frac{\mathbf{1}^T(x(w)_{1:T} - cx(w)_{0:T-1})}{T(1-c)} = \frac{\mathbf{1}^T A(c)w}{(1-c)T}$$

We will substitute $\theta^*(c, w)$ into f to obtain the first minimization. Let:

$$B = I - \frac{1}{T}\mathbf{1}\mathbf{1}^T$$

Then substituting θ^* back :

$$\begin{aligned} f_1(\mathbf{w}, a, c) &= \frac{\ln(a)}{2} + \frac{\left\| A(c)\mathbf{w} - \left[\frac{\mathbf{1}^T A(c)\mathbf{w}}{(1-c)T}\right](1-c)\mathbf{1} \right\|^2}{2Ta} + \gamma c \\ \leftrightarrow f_1(\mathbf{w}, a, c) &= \frac{\ln(a)}{2} + \frac{\left\| A(c)\mathbf{w} - \left[\frac{\mathbf{1}\mathbf{1}^T A(c)\mathbf{w}}{T}\right] \right\|^2}{2Ta} + \gamma c \\ \leftrightarrow f_1(\mathbf{w}, a, c) &= \frac{\ln(a)}{2} + \frac{\left\| \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{T}\right) A(c)\mathbf{w} \right\|^2}{2Ta} + \gamma c \\ \leftrightarrow f_1(\mathbf{w}, a, c) &= \frac{\ln(a)}{2} + \frac{\|\mathbf{B}(x(\mathbf{w})_{1:T} - cx(\mathbf{w})_{0:T-1})\|^2}{2Ta} + \gamma c \end{aligned}$$

Where we used the fact:

$$A(c)w - \theta^*(1-c)\mathbf{1} = BA(c)w$$

Then:

$$\leftrightarrow f_1(w,a,c) = \frac{\ln(a)}{2} + \frac{\|B(x(w)_{1:T} - cx(w)_{0:T-1})\|^2}{2Ta} + \gamma c$$

The function f_1 is differentiated with respect to c where $\partial_c f_1 = 0$ so that:

$$\frac{\partial}{\partial c}\left[\frac{\ln(a)}{2} + \frac{\|BA(c)w\|^2}{2Ta} + \gamma c\right] = 0$$
$$\leftrightarrow \gamma + \frac{1}{2Ta}\frac{\partial}{\partial c}[\|B(x(w)_{1:T} - cx(w)_{0:T-1})\|^2] = 0$$

Now continue with the second minimization in respect to c:

$$f_2(w,a) = \min_c f_1(w,a,c) = \min_{c,\theta} f(w,a,c,\theta)$$

The function f_1 is differentiated with respect to c where $\partial_c f_1 = 0$ so that:

$$\frac{\partial}{\partial c}\left[\frac{\ln(a)}{2} + \frac{\|BA(c)w\|^2}{2Ta} + \gamma c\right] = 0$$
$$\leftrightarrow \gamma + \frac{1}{2Ta}\frac{\partial}{\partial c}[\|B(x(w)_{1:T} - cx(w)_{0:T-1})\|^2] = 0$$

Let

$$u = B(x(w)_{1:T} - cx(w)_{0:T-1})$$
$$\leftrightarrow u = Bx(w)_{1:T} - cBx(w)_{0:T-1}$$

For simplification, let $b_{1:T}(w) = Bx(w)_{1:T}$ and $b_{0:T-1}(w) = Bx(w)_{0:T-1}$, then:

$$u = b_{1:T}(w) - cb_{0:T-1}(w)$$
$$\leftrightarrow \partial u = -b_{0:T-1}(w)\partial c$$

also let

$$z = \|u\|^2 = u^T u$$
$$\leftrightarrow \partial z = \partial u^T u + u^T \partial u$$
$$\leftrightarrow \partial z = \partial u^T u + \partial u^T u = 2\partial u^T u$$

then

$$\leftrightarrow \partial z = 2(-b_{0:T-1}(w)\partial c)^T u$$

So that:

$$b_1(w) = Bx_{1:T}, \qquad b_0(w) = Bx_{0:T-1}$$

$$\begin{aligned}\leftrightarrow \partial z &= -2b_{0:T-1}^T(w)u\,\partial c \\ \leftrightarrow \frac{\partial z}{\partial c} &= -2b_{0:T-1}^T\left(b_{1:T}(w) - cb_{0:T-1}(w)\right)\end{aligned}$$

so that

$$\begin{aligned}\gamma + \frac{1}{2Ta}\frac{\partial}{\partial c}\|B(x(w)_{1:T} - cx(w)_{0:T-1})\|^2 &= 0 \\ \leftrightarrow \gamma + \frac{1}{2Ta}\frac{\partial}{\partial c}\|u\|^2 &= 0 \\ \leftrightarrow \gamma + \frac{-1}{2Ta}2b_{0:T-1}^T\left(b_{1:T}(w) - cb_{0:T-1}(w)\right) &= 0 \\ \leftrightarrow -\frac{1}{Ta}b_{0:T-1}^T\left(b_{1:T}(w) - cb_{0:T-1}(w)\right) &= -\gamma \\ \leftrightarrow b_{0:T-1}^T b_{1:T}(w) - cb_{0:T-1}^T b_{0:T-1}(w) &= \gamma Ta \\ \leftrightarrow cb_{0:T-1}^T b_{0:T-1}(w) &= -\gamma Ta + b_{0:T-1}^T b_{1:T}(w)\end{aligned}$$

Note that $b_{0:T-1}^T b_{0:T-1}(w) = \|b_{0:T-1}(w)\|^2$. If $b_{0:T-1}(w) = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{T-1} \end{bmatrix}$, then

$$\begin{aligned}\|b_{0:T-1}(w)\|^2 &= \left(\sqrt{b_0^2 + b_1^2 + \cdots + b_{T-1}^2}\right)^2 \\ \|b_{0:T-1}(w)\|^2 &= b_0^2 + b_1^2 + \cdots + b_{T-1}^2 \\ \|b_{0:T-1}(w)\|^2 &= \begin{bmatrix} b_0 & b_1 & \dots & b_{T-1} \end{bmatrix}\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{T-1} \end{bmatrix} \\ \|b_{0:T-1}(w)\|^2 &= b_{0:T-1}^T b_{0:T-1}(w)\end{aligned}$$

Therefore, the previous equation is equivalent to

$$\begin{aligned}c\|b_{0:T-1}(w)\|^2 &= b_{0:T-1}^T b_{1:T}(w) - \gamma Ta \\ \leftrightarrow c^* &= \frac{b_{0:T-1}^T b_{1:T}(w) - \gamma Ta}{\|b_{0:T-1}(w)\|^2}\end{aligned}$$

then this means that:

$$BA(c)w = b_1(w) - cb_0(w)$$

In reduced form, the objective is:

$$f_1(w, a, c) = \frac{1}{2}\ln(a) + \frac{1}{2Ta}\|b_1 - cb_0\|^2 + \gamma c$$

4.1.1 $\gamma = 0$

Now we minimize over c. The case of $\gamma = 0$:

$$\frac{\partial}{\partial c}\|b_1 - cb_0\|^2 = -2b_0^T(b_1 - cb_0)$$

Set to zero:

$$b_0^T(b_1 - cb_0) = 0$$

Solve:

$$c^* = \frac{b_0^T b_1}{\|b_0\|^2}$$

Minimize over a:

$$f_2(w, a) = \frac{1}{2}\ln(a) + \frac{1}{2Ta}\|b_1 - c^* b_0\|^2$$

Derivative:

$$\frac{\partial f_2}{\partial a} = \frac{1}{2a} - \frac{1}{2Ta^2}\|b_1 - c^* b_0\|^2$$

Set to zero:

$$\frac{1}{2a} = \frac{1}{2Ta^2}\|b_1 - c^* b_0\|^2$$

Multiply both sides by $2a^2$:

$$a = \frac{1}{T}\|b_1 - c^* b_0\|^2$$

The final reduced objective is substituting for a^*

$$f_3(\mathbf{w}) = f_2(a^*, \mathbf{w}) = \frac{\ln(a^*)}{2} + \frac{\left\|b_{1:T}(\mathbf{w}) - \frac{b_{0:T-1}^\top b_{1:T}(\mathbf{w})}{\|b_{0:T-1}(\mathbf{w})\|^2} b_{0:T-1}(\mathbf{w})\right\|^2}{2T\frac{1}{T}\|b_{1:T}(\mathbf{w}) - c^*(\mathbf{w})b_{0:T-1}(\mathbf{w})\|^2} = \frac{\ln(a^*) + 1}{2}$$

4.1.2 $\gamma > 0$

Solving when the penalized part is not zero is trickier. Take the derivative as usual:

$$
\begin{aligned}
f_2(\mathbf{w}, a) &= \frac{\ln(a)}{2} + \frac{\left\| \mathbf{b}_1 - \mathbf{b}_0 \frac{\left(\mathbf{b}_0^T \mathbf{b}_1 - \gamma T a\right)}{\|\mathbf{b}_0\|^2} \right\|^2}{2Ta} + \gamma \left(\frac{\mathbf{b}_0^T \mathbf{b}_1 - \gamma T a}{\|\mathbf{b}_0\|^2} \right) \\
&= \frac{\ln(a)}{2} + \frac{1}{2Ta} \left\| \mathbf{b}_1 - \frac{\mathbf{b}_0 \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} + \frac{\gamma T a \mathbf{b}_0}{\|\mathbf{b}_0\|^2} \right\|^2 + \frac{\gamma \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} - \frac{\gamma^2 T a}{\|\mathbf{b}_0\|^2} \\
&= \frac{\ln(a)}{2} + \frac{1}{2Ta} \left(\mathbf{b}_1 - \frac{\mathbf{b}_0 \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} + \frac{\gamma T a \mathbf{b}_0}{\|\mathbf{b}_0\|^2} \right)^T \left(\mathbf{b}_1 - \frac{\mathbf{b}_0 \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} + \frac{\gamma T a \mathbf{b}_0}{\|\mathbf{b}_0\|^2} \right) \\
&\quad + \frac{\gamma \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} - \frac{\gamma^2 T a}{\|\mathbf{b}_0\|^2} \\
&= \frac{\ln(a)}{2} + \frac{1}{2Ta} \left(\mathbf{b}_1^T - \frac{\mathbf{b}_1^T \mathbf{b}_0 \mathbf{b}_0^T}{\|\mathbf{b}_0\|^2} + \frac{\gamma T a \mathbf{b}_0^T}{\|\mathbf{b}_0\|^2} \right) \left(\mathbf{b}_1 - \frac{\mathbf{b}_0 \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} + \frac{\gamma T a \mathbf{b}_0}{\|\mathbf{b}_0\|^2} \right) + \frac{\gamma \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} \\
&\quad - \frac{\gamma^2 T a}{\|\mathbf{b}_0\|^2} \\
&= \frac{\ln(a)}{2} \\
&\quad + \frac{1}{2Ta} \left(\|\mathbf{b}_1\|^2 - \frac{(\mathbf{b}_0^T \mathbf{b}_1)^2}{\|\mathbf{b}_0\|^2} + \frac{\gamma T a \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} - \frac{(\mathbf{b}_0^T \mathbf{b}_1)^2}{\|\mathbf{b}_0\|^2} + \frac{(\mathbf{b}_0^T \mathbf{b}_1)^2}{\|\mathbf{b}_0\|^2} \right. \\
&\quad \left. - \frac{\gamma T a \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} + \frac{\gamma T a \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} - \frac{\gamma T a \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} + \frac{\gamma^2 T^2 a^2}{\|\mathbf{b}_0\|^2} \right) + \frac{\gamma \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} - \frac{\gamma^2 T a}{\|\mathbf{b}_0\|^2} \\
&= \frac{\ln(a)}{2} + \frac{1}{2Ta} \left(\|\mathbf{b}_1\|^2 - \frac{2(\mathbf{b}_0^T \mathbf{b}_1)^2}{\|\mathbf{b}_0\|^2} + \frac{(\mathbf{b}_0^T \mathbf{b}_1)^2}{\|\mathbf{b}_0\|^2} + \frac{\gamma^2 T^2 a^2}{\|\mathbf{b}_0\|^2} \right) + \frac{\gamma \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} - \frac{\gamma^2 T a}{\|\mathbf{b}_0\|^2} \\
&= \frac{\ln(a)}{2} + \frac{\|\mathbf{b}_1\|^2}{2Ta} - \frac{(\mathbf{b}_0^T \mathbf{b}_1)^2}{Ta\|\mathbf{b}_0\|^2} + \frac{(\mathbf{b}_0^T \mathbf{b}_1)^2}{2Ta\|\mathbf{b}_0\|^2} + \frac{\gamma^2 T a}{2\|\mathbf{b}_0\|^2} + \frac{\gamma \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} - \frac{\gamma^2 T a}{\|\mathbf{b}_0\|^2} \\
&= \frac{\ln(a)}{2} + \frac{\|\mathbf{b}_1\|^2}{2Ta} + \frac{\gamma \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2} + \left(\frac{1}{2} - 1 \right) \frac{(\mathbf{b}_0^T \mathbf{b}_1)^2}{Ta\|\mathbf{b}_0\|^2} + \left(\frac{1}{2} - 1 \right) \frac{\gamma^2 T a}{\|\mathbf{b}_0\|^2} \\
&= \frac{\ln(a)}{2} + \frac{\|\mathbf{b}_1\|^2}{2Ta} - \frac{(\mathbf{b}_0^T \mathbf{b}_1)^2}{2Ta\|\mathbf{b}_0\|^2} - \frac{\gamma^2 T a}{2\|\mathbf{b}_0\|^2} + \frac{\gamma \mathbf{b}_0^T \mathbf{b}_1}{\|\mathbf{b}_0\|^2}
\end{aligned}
$$

Set everything to zero, so we can find that:

$$\frac{\partial}{\partial a} f_2(\mathbf{w}, a) = \frac{\partial}{\partial a}\left(\frac{\ln(a)}{2} + \frac{\|\mathbf{b}_1\|^2}{2Ta} - \frac{(\mathbf{b}_0^T\mathbf{b}_1)^2}{2Ta\|\mathbf{b}_0\|^2} - \frac{\gamma^2 Ta}{2\|\mathbf{b}_0\|^2} + \frac{\gamma\mathbf{b}_0^T\mathbf{b}_1}{\|\mathbf{b}_0\|^2}\right) = 0$$

$$\begin{aligned}
&\leftrightarrow \frac{1}{2a} - \frac{\|\mathbf{b}_1\|^2}{2Ta^2} + \frac{(\mathbf{b}_0^T\mathbf{b}_1)^2}{2Ta^2\|\mathbf{b}_0\|^2} - \frac{\gamma^2 T}{2\|\mathbf{b}_0\|^2} = 0 \\
&\leftrightarrow aT\|\mathbf{b}_0\|^2 - \|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 + (\mathbf{b}_0^T\mathbf{b}_1)^2 - \gamma^2 T^2 a^2 = 0 \\
&\leftrightarrow \gamma^2 T^2 a^2 - aT\|\mathbf{b}_0\|^2 + \|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2 = 0
\end{aligned}$$

This reduces into a simple quadratuc equation, by which we solve:

$$a^* = \frac{\|b_0\|^2 - \sqrt{\|b_0\|^4 - 4\gamma^2(\|b_0\|^2\|b_1\|^2 - (b_0^T b_1)^2)}}{2T\gamma^2}$$

This can be solved via:

$$\begin{aligned}
&\leftrightarrow a^* = \frac{T\|\mathbf{b}_0\|^2 \pm T\sqrt{\|\mathbf{b}_0\|^4 - 4\gamma^2(\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2)}}{2\gamma^2 T^2} \\
&\leftrightarrow a^* = \frac{\|\mathbf{b}_0\|^2 \pm \sqrt{\|\mathbf{b}_0\|^4 - 4\gamma^2(\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2)}}{2\gamma^2 T} \\
&\leftrightarrow a^* = \frac{1}{2\gamma^2 T}\left(\|\mathbf{b}_0\|^2 \pm \sqrt{\|\mathbf{b}_0\|^4 - 4\gamma^2(\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2)}\right)
\end{aligned}$$

Then, a^* is chosen as:

$$a^* = \frac{1}{2\gamma^2 T}\left(\|\mathbf{b}_0\|^2 - \sqrt{\|\mathbf{b}_0\|^4 - 4\gamma^2(\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2)}\right)$$

where:

$$\begin{aligned}
&\|\mathbf{b}_0\|^4 - 4\gamma^2(\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2) \geq 0 \\
\leftrightarrow &- 4\gamma^2(\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2) \geq -\|\mathbf{b}_0\|^4 \\
\leftrightarrow &4\gamma^2(\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2) \leq \|\mathbf{b}_0\|^4 \\
\leftrightarrow &\gamma^2 \leq \frac{\|\mathbf{b}_0\|^4}{4(\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2)} \\
\leftrightarrow &\gamma \leq \frac{1}{2}\sqrt{\frac{\|\mathbf{b}_0\|^4}{\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2}} \\
\leftrightarrow &0 < \gamma \leq \frac{1}{2}\sqrt{\frac{\|\mathbf{b}_0\|^4}{\|\mathbf{b}_0\|^2\|\mathbf{b}_1\|^2 - (\mathbf{b}_0^T\mathbf{b}_1)^2}}
\end{aligned}$$

so that the value of f_3 is:

$$f_3(\mathbf{w}) = f_2(a^*, \mathbf{w}) = \frac{\ln(a^*)}{2} + \frac{\|\mathbf{b}_1\|^2}{2Ta^*} - \frac{(\mathbf{b}_0^T\mathbf{b}_1)^2}{2Ta^*\|\mathbf{b}_0\|^2} - \frac{\gamma^2 Ta^*}{2\|\mathbf{b}_0\|^2} + \frac{\gamma\mathbf{b}_0^T\mathbf{b}_1}{\|\mathbf{b}_0\|^2}$$

Then :

$$f_3(w) = \frac{1}{2}\ln(a^*) + \frac{\|b_1\|^2}{2Ta^*} - \frac{(b_0^T b_1)^2}{2Ta^*\|b_0\|^2} - \frac{Ta^*\gamma^2}{2\|b_0\|^2} + \frac{\gamma b_0^T b_1}{\|b_0\|^2}$$

The final form is:

$$\min_{\|w\|_1=1,\ \|w\|_0\leq\eta} f_3(w)$$