Sentiment-Adjusted Black-Litterman Optimizer

Mark's Quant
Bereich: Quantitative Finance & Data Science

11. Februar 2026

Zusammenfassung

Dieses Whitepaper präsentiert eine produktionsreife Implementierung des Black-Litterman (BL)-Modells, ergänzt durch eine Stimmungsanalyse von FinBERT, einem vortrainierten Sprachmodell, das auf Finanztexte spezialisiert ist. Ausgehend von Attilio Meuccis (2008) marktbasierter Formulierung des BL-Modells befassen wir uns mit dem Schätzungsrisiko bei der Portfoliooptimierung, indem wir Gleichgewichtsvorannahmen mit Anlegermeinungen kombinieren, die aus NLP-gesteuerten Stimmungswerten abgeleitet werden. Wir beschreiben detailliert die mathematischen Grundlagen, einschließlich der Ableitung der posterioren Verteilung, der Kalibrierung der Ansichten über Volatilitätsskalierung und Unsicherheitsgewichtung sowie der Integration mit Echtzeit-Nachrichten. Das Framework ist in Python implementiert, wodurch numerische Stabilität, Typ-Hinting und Effizienz gewährleistet sind. Empirische Beispiele zeigen, wie dieser Ansatz zu intuitiven Allokationen führt, Ecklösungen vermeidet und für die Vermögensverwaltung unter Solvency II/IFRS 17 geeignet ist. Der Code ist auf GitHub verfügbar, um die Reproduzierbarkeit zu gewährleisten.

1 Einleitung

Die traditionelle Mean-Variance-Optimierung (Markowitz, 1952) leidet unter einem Schätzungsrisiko, was zu instabilen Portfolios mit extremen Gewichtungen führt. Das Black-Litterman-Modell (Black & Litterman, 1992) mildert dies, indem es die Ansichten der Anleger in einen bayesschen Rahmen einbezieht, ausgehend von einer stabilen Prior-Verteilung (z. B. CAPM-Gleichgewicht). Die Erweiterung von Attilio Meucci (2008) verfeinert dies, indem sie Ansichten zu Marktrenditen anstelle von Parametern formuliert und so das Grenzverhalten und die Sparsamkeit verbessert. In vorliegendem Whitepaper erfolgt eine Erweiterung von BL durch die Integration der Sentimentanalyse aus FinBERT (Araci, 2019). Hierbei handelt es sich um ein BERT-basiertes Modell, das auf einem Finanzkorpus basiert. Die Sentiment-Werte aus Nachrichtenartikeln werden in BL-Meinungen kalibriert, wodurch die Herausforderung der Übersetzung unstrukturierter Daten in quantitative Eingaben bewältigt wird. Unsere Umsetzung konzentriert sich auf:

- Mathematische Genauigkeit: Verwendung von Meuccis Marktformulierung für die posteriore Berechnung.
- Sentimentkalibrierung: Volatilitätsgebundene Anpassungen mit Volumen- und Konsistenzgewichtung.
- Produktionsreife: Python-Code mit Typ-Hinweisen, Fehlerbehandlung und Visualisierung über Plotly.

Wir konzentrieren uns auf Anwendungen im Bereich Vermögens- und Risikomanagement und legen dabei den Schwerpunkt auf Funktionen wie Kovarianzschrumpfung (Ledoit & Wolf, 2004) und Backtesting für die Attraktivität von Personalvermittlern. Das Framework wurde anhand historischer Daten validiert und entspricht den Prinzipien von Meucci (2005) zu „Risiko und Vermögensallokation".

2 Mathematische Grundlagen

2.1 Das Black-Litterman Modell

Betrachten wir einen Markt mit N Vermögenswerten mit Renditen $X \sim \mathcal{N}(\mu, \Sigma)$, wobei μ der erwartete Renditevektor und Σ die Kovarianzmatrix ist. Das Schätzungsrisiko macht μ ungewiss; BL modelliert es als $\mu \sim \mathcal{N}(\pi, \tau\Sigma)$, wobei π die Prior-Verteilung (z.B. Gleichgewichtsrenditen) und τ die Prior-Ungewissheit (typischerweise 0.01˘0.05) ist.

Gleichgewichts-Prior

Das CAPM-Gleichgewicht setzt $\pi = \delta\Sigma w_m$, wobei w_m die Marktkapitalisierungsgewichte und δ die Risikoaversion (z. B. 2.5) sind. Dies führt zu einer umgekehrten Optimierung der beobachteten Gewichte:

$$\pi = \delta\Sigma w_m$$

(Black & Litterman, 1992; Meucci, 2005).

Ansichten der Anleger

Die Ansichten sind linear: $P\mu \sim \mathcal{N}(Q, \Omega)$, wobei $P(K\text{Ö}N)$ die Auswahlmatrix, $Q(K\text{Ö}1)$ die Rendite der Ansicht und $\Omega(K\text{Ö}K)$ die Unsicherheit ist.

In der ursprünglichen Formulierung (He & Litterman, 2002) lautet die Posterior-Verteilung:

$$\mu_{BL} = \pi + \tau\Sigma P^T(\tau P\Sigma P^T + \Omega)^{-1}(Q - P\pi)$$

$$\Sigma_{BL} = (1+\tau)\Sigma - \tau^2\Sigma P^T(\tau P\Sigma P^T + \Omega)^{-1}P\Sigma$$

(Meucci, 2008, Gleichungen 20–21).

Marktbasierte Formulierung (empfohlen)

Meucci (2008, Abschnitt 3) formuliert die Ansichten zu den Erträgen X neu: $PX \sim \mathcal{N}(Q, \Omega)$. Die A-posteriori-Verteilung vereinfacht sich (kein τ):

$$\mu_{BL}^m = \pi + \Sigma P^T(P\Sigma P^T + \Omega)^{-1}(Q - P\pi)$$

$$\Sigma_{BL}^m = \Sigma - \Sigma P^T (P\Sigma P^T + \Omega)^{-1} P\Sigma$$

(Gleichungen 32–33). Dies gewährleistet korrekte Grenzen: Wenn $\Omega \to \infty$ (geringe Konfidenz), dann ist die Posterior-Verteilung $\to$ Prior-Verteilung; wenn $\Omega \to 0$ (hohe Konfidenz), dann ist die Posterior-Verteilung $\to$ bedingte Verteilung.

Ableitung (Anhang A):
Aus der Bayes-Regel folgt: $f(X|Q) \propto f(Q|X)f(X)$. Durch Vervollständigung des Quadrats ergibt sich das Obige.

Unsicherheitsmatrix Ω
Meucci empfiehlt $\Omega = \frac{1}{c} P\Sigma P^T$, wobei c die Konfidenz ist (höherer $c \to$ niedrigerer Ω). Für relative Konfidenzen: $\Omega = \frac{1}{c} \operatorname{diag}(u) P\Sigma P^T \operatorname{diag}(u)$, wobei u der relative Unsicherheitsvektor ist.

2.2 Optimierung

Die posterioren Parameter fließen in die Mittelwert-Varianz ein: $\max_w w^T \mu_{BL} - \frac{\delta}{2} w^T \Sigma_{BL} w$, vorbehaltlich bestimmter Einschränkungen (z. B. keine negativen Werte: $w \geq 0, \sum w = 1$).

3 Kalibrierung der Stimmung: Integration von FinBERT-Ansichten

FinBERT (Araci, 2019) extrahiert die Stimmung aus Finanztexten und bewertet sie mit [-1, +1]. Wir kalibrieren diese in BL-Ansichten und schlagen damit eine Brücke zwischen NLP und quantitativer Finanzwissenschaft.

3.1 Konstruktion der Ansichten

Für absolute Ansichten zu Vermögenswert i: $P[k, i] = 1$.
Ansichtsvektor Q: Volatilitätsskaliert: $Q[k] = \pi[i] + s[k] \cdot \sigma[i] \cdot \alpha$, wobei $s[k]$ der Sentiment-Mittelwert ist, $\sigma[i] = \sqrt{\Sigma_{i,i}}$ und α (Skalierung, Standardwert 0.02) den Einfluss steuert.
Ökonomische Begründung: Vermögenswerte mit hoher Volatilität erhalten größere Anpassungen (z. B. $\sigma = 30\%$, $s = 0,5$: +0,3 % Anpassung). Backtest α für Realismus.

3.2 Unsicherheitsmatrix Ω

$\Omega[k, k] = \beta + w_v \cdot (\frac{1}{\sqrt{n}}) + w_c \cdot v^2$, wobei β die Basisunsicherheit (0.0001), n die Anzahl der Nachrichten, v die Standardabweichung der Stimmung und $w_v = w_c = 0.5$ sind. Interpretation: Hohes Volumengeringe Varianz $\to$ niedriger Ω (hohe Zuverlässigkeit). Bestraft verrauschte Signale.

3.3 Filterung und Validierung

Filtern Sie schwache Ansichten: $|s| < \theta$ (z. B. 0.15), min. Nachrichten = 3. Validieren: Überprüfen Sie PSD ($\Omega > 0$), Grenzen ($|Q - \pi| < 10\%$).
Integration mit NewsAPI: Texte abrufen, mit FinBERT bewerten: $s = \text{Mittelwert(scores)}, v = \text{Standardabweichung(scores)}$.

4 Python-Implementierung

4.1 Core Klassen

BlackLittermanModel: Berechnet das Gleichgewicht π, posterior über Meucci. Typgeprüfte, stabile Umkehrungen (`np.linalg.inv` mit Konditionierung).

```
def compute_posterior(self, P: np.ndarray, Q: np.ndarray, Omega: np.ndarray) -> Tuple[np.ndarray,
    """Implements Meucci's Equations (32-33)."""
    M = P @ self.Sigma @ P.T + Omega
    M_inv = np.linalg.inv(M)
    adjustment = Q - P @ self.pi
    mu_BL = self.pi + self.Sigma @ P.T @ M_inv @ adjustment
    Sigma_BL = self.Sigma - self.Sigma @ P.T @ M_inv @ P @ self.Sigma
    return mu_BL, Sigma_BL
```

ViewGenerator: Kalibriert die Stimmung auf (P, Q, Ω). Verwendet die Datenklasse `SentimentData`.
Integration: Aktualisiert Skripte wie `bl_data_views_1.py` mit gleichzeitiger Abfrage (ThreadPoolExecutor) und effizienten Plotly-Grenzen.

4.2 Numerische Stabilität

Schrumpfung: $\Sigma = (1 - \lambda)\hat{\Sigma} + \lambda\, \text{diag}(\hat{\Sigma})$ (Ledoit-Wolf).
PSD-Projektion: Wenn Eigenwerte < 0, auf $1e - 8$ setzen.
Fehlerbehandlung: API-Fehler werden mit exponentiellem Backoff wiederholt.
Enthalten ist unter anderem Multithreading für Nachrichten, vektorisierte NumPy-Operationen für mehr Geschwindigkeit.

5 Praktische Anwendung und Beispiele

5.1 Arbeitsablauf

1. Preise (`yfinance`) und Nachrichten (NewsAPI) abrufen.
2. Stimmung berechnen (FinBERT-Pipeline).
3. Ansichten (Views) generieren.
4. Posterior berechnen.

5. Optimieren (SciPy minimieren mit Einschränkungen).
6. Visualisieren: Plotly-Effizienzgrenze.

Beispiel: Für Sektoren (XLK, XLE usw.) beeinflusst die Stimmung die Gewichtung (z.B. positive Stimmung im Technologiesektor → höhere Allokation).

5.2 Backtesting

Monatlich: Sharpe-Ratio vs. Benchmark vergleichen. Glättung der Stimmung: $s_t = 0,7s_{t-1} + 0,3s_{\text{new}}$.

6 Grenzen des Modells und Ausblick

Obwohl die hier präsentierte Sentiment-adjustierte Black-Litterman-Implementierung (basierend auf Meucci, 2008) Schätzungsrisiken durch Bayesianische Integration von Priors (Gleichgewichtsrenditen $\pi = \delta\Sigma w_m$) und Views ($P\mu\tilde{}N(Q,\Omega)$) mildert, unterliegt sie typischen Annahmen der Mean-Variance-Optimierung. Die Renditen werden als multivariat normalverteilt modelliert ($X\tilde{}N(\mu_{BL},\Sigma_{BL})$), was in realen Märkten oft verletzt wird.

Eine zentrale Limitation sind Fat Tails: Empirische Renditeverteilungen weisen höhere Kurtosis auf als die Normalverteilung, was extreme Ereignisse (z.B. Marktabstürze) unterschätzt. Mathematisch führt dies zu einer Unterbewertung des Tail-Risikos in der posterioren Verteilung:

$$\mu_{BL} = \pi + \Sigma P^T(P\Sigma P^T + \Omega)^{-1}(Q - P\pi),$$

da die Kovarianz Σ keine leptokurtischen Effekte (wie stark die Renditeverteilung von Finanzanlagen „fette“ oder „schwere“ Ausläufer (Fat Tails) aufweist) berücksichtigt (vgl. Meucci, 2005, Kap. 4 zu Risiko und Vermögensallokation). Dies kann zu instabilen Allokationen in volatilen Phasen führen.

Eine weitere Einschränkung sind Regime-Wechsel, bei denen Korrelationen gegen 1 konvergieren (z.B. in Finanzkrisen, wo Diversifikation versagt). Die statische Kovarianzschrumpfung (Ledoit & Wolf, 2004) ignoriert dynamische Regime-Shifts, was die Unsicherheit Ω in den Views nicht ausreichend anpasst und zu überoptimistischen Portfolios führen kann.

Für eine Erweiterung empfehle ich die Integration von Copulas zur Modellierung nichtnormaler Abhängigkeiten oder robuste Optimierung (z.B. via Elliptische Verteilungen), um numerische Stabilität zu gewährleisten. Dies würde das Framework für Solvency II-konforme Stress-Tests attraktiver machen und könnte durch Monte-Carlo-Simulationen implementiert werden - ein Feature, das in Produktionsumgebungen (mit Multithreading) beeindruckend wirkt. Zukünftige Arbeit könnte FinBERT um Regime-Erkennung erweitern, um Views dynamisch anzupassen.

7 Fazit

Dieses Framework kombiniert die bayessche Genauigkeit von BL mit der NLP-Leistung von FinBERT und liefert stabile, ansichtsgesteuerte Allokationen. Zukünftige Erweiterungen: Nicht-normale Märkte (Meucci, 2006), Ansichten mehrerer Benutzer. GitHub-Repo: [Link].

Literatur

[1] Araci, D. (2019). *FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models*. arXiv:1908.10063.

[2] Black, F., & Litterman, R. (1992). Global Portfolio Optimization. *Financial Analysts Journal*, 48(5), 28–43.

[3] He, G., & Litterman, R. (2002). *The Intuition Behind Black-Litterman Model Portfolios*. Goldman Sachs.

[4] Ledoit, O., & Wolf, M. (2004). Honey, I Shrunk the Sample Covariance Matrix. *Journal of Portfolio Management*, 30(4), 110–119.

[5] Markowitz, H. (1952). Portfolio Selection. *Journal of Finance*, 7(1), 77–91.

[6] Meucci, A. (2005). *Risk and Asset Allocation*. Springer.

[7] Meucci, A. (2008). *The Black-Litterman Approach: Original Model and Extensions*. SSRN:1117574.

Appendix A: Posterior Ableitung

From Bayes: $f(X|Q) \propto \exp\Big(-\frac{1}{2}(X-\pi)^T\Sigma^{-1}(X-\pi) - \frac{1}{2}(Q-PX)^T\Omega^{-1}(Q-PX)\Big)$.
Precision: $\Sigma_{BL}^{-1} = \Sigma^{-1} + P^T\Omega^{-1}P$.
Mean: $\mu_{BL} = \Sigma_{BL}(\Sigma^{-1}\pi + P^T\Omega^{-1}Q)$.