Social Upheaval Composite Index: Mathematical Framework

Core Design Principles

1. Multi-Dimensional Approach

Instead of single subjective score, measure distinct types of upheaval that could influence cultural production.

2. Standardization Across Time

Each component normalized to 0-100 scale to allow comparison across decades with different baseline conditions.

3. Weighted Aggregation

Different types of upheaval may have different impacts on cultural anxiety and film production.

Component Dimensions

A. Political Violence & Instability (25% weight)

Rationale: Direct threats to social order create immediate cultural anxiety

Measurable Indicators:

- Political assassinations (presidents, major politicians, activists)
- Domestic terrorist attacks with political motivation
- Major riots/civil unrest (number and severity)
- Political protests (frequency and size)
- Government instability (resignations, impeachments)

Scoring Method:

```
def calculate_political_violence_score(decade_data):
    score = 0

    # High-impact events (weighted heavily)
    score += decade_data['assassinations_major'] * 20
    score += decade_data['terrorist_attacks_domestic'] * 15
    score += decade_data['riots_major'] * 10

    # Medium-impact events
    score += decade_data['protests_large'] * 5
    score += decade_data['government_crises'] * 8

    # Cap at 100, normalize
    return min(score, 100)
```

B. Institutional Trust Erosion (20% weight)

Rationale: Loss of faith in institutions creates societal paranoia themes

Measurable Indicators:

- Major political scandals (Watergate-level)
- Supreme Court controversial decisions
- Military/intelligence failures or scandals
- Media credibility crises
- Electoral integrity questions

Scoring Method:

```
def calculate_institutional_trust_score(decade_data):
    score = 0

    # Major scandals with lasting impact
    score += decade_data['major_scandals'] * 25
    score += decade_data['supreme_court_controversial'] * 10
```

```
    score += decade_data['intelligence_scandals'] * 15
    score += decade_data['electoral_controversies'] * 12

    return min(score, 100)
```

C. Economic Stress & Inequality (15% weight)

Rationale: Economic anxiety drives demand for films exploring systemic problems

Measurable Indicators:

- Recession severity and duration
- Unemployment peaks
- Income inequality measures (Gini coefficient changes)
- Major corporate/financial scandals
- Housing/cost of living crises

Scoring Method:

```
def calculate_economic_stress_score(decade_data):
    score = 0

    # Recession impact
    recession_severity = decade_data['recession_months'] * decade_data['une
mployment_peak']
    score += min(recession_severity, 40)

    # Inequality changes
    gini_change = decade_data['gini_coefficient_change'] * 100
    score += max(gini_change, 0) * 30  # Only increases count

    # Financial scandals
    score += decade_data['major_financial_scandals'] * 15

    return min(score, 100)
```

D. External Threats & Conflicts (20% weight)

Rationale: External dangers create paranoid/thriller cultural themes

Measurable Indicators:

- War involvement (duration, casualties, controversy)
- International terrorist threats
- Cold War tensions/nuclear fears
- Foreign interference in elections
- Pandemic/health crises

Scoring Method:

```
def calculate_external_threats_score(decade_data):
    score = 0

    # War involvement
    war_impact = (decade_data['war_years'] * decade_data['casualty_rate'] *
                  decade_data['controversy_factor'])
    score += min(war_impact, 35)

    # Terrorism/security threats
    score += decade_data['terror_threat_level'] * 10
    score += decade_data['foreign_interference'] * 15

    # Health/pandemic crises
    score += decade_data['pandemic_severity'] * 20

    return min(score, 100)
```

E. Social Fragmentation (20% weight)

Rationale: Division and polarization drive demand for films exploring "us vs them" themes

Measurable Indicators:

- Political polarization measures
- Racial/ethnic tensions and incidents
- Generational conflicts
- Regional divisions
- Information/media fragmentation

Scoring Method:

```
def calculate_social_fragmentation_score(decade_data):
    score = 0

    # Polarization metrics
    score += decade_data['political_polarization_index'] * 25
    score += decade_data['racial_tension_incidents'] * 15

    # Information environment
    score += decade_data['media_fragmentation_index'] * 20
    score += decade_data['disinformation_prevalence'] * 15

    # Regional/cultural divisions
    score += decade_data['cultural_conflict_intensity'] * 25

    return min(score, 100)
```

Master Composite Index Formula

Weighted Aggregation

```
def calculate_composite_upheaval_index(decade_data, weights=None):

    if weights is None:
        weights = {
            'political_violence': 0.25,
            'institutional_trust': 0.20,
```

```
        'economic_stress': 0.15,
        'external_threats': 0.20,
        'social_fragmentation': 0.20
    }

    # Calculate component scores
    pv_score = calculate_political_violence_score(decade_data)
    it_score = calculate_institutional_trust_score(decade_data)
    es_score = calculate_economic_stress_score(decade_data)
    et_score = calculate_external_threats_score(decade_data)
    sf_score = calculate_social_fragmentation_score(decade_data)

    # Weighted composite
    composite_score = (
        pv_score * weights['political_violence'] +
        it_score * weights['institutional_trust'] +
        es_score * weights['economic_stress'] +
        et_score * weights['external_threats'] +
        sf_score * weights['social_fragmentation']
    )

    return {
        'composite_score': composite_score,
        'components': {
            'political_violence': pv_score,
            'institutional_trust': it_score,
            'economic_stress': es_score,
            'external_threats': et_score,
            'social_fragmentation': sf_score
        }
    }
```

Alternative Aggregation Methods

1. Multiplicative Model (Crisis Amplification)

```
# Assumes components amplify each other during true upheaval
def multiplicative_upheaval_index(components):
    base_score = sum(components.values()) / len(components)
    amplification = 1.0

    # If multiple components are high, amplify the effect
    high_components = sum(1 for score in components.values() if score > 70)
    if high_components >= 3:
        amplification = 1.5
    elif high_components >= 2:
        amplification = 1.2

    return min(base_score * amplification, 100)
```

2. Peak-Sensitive Model

```
# Emphasizes the highest single component (worst crisis dominates)
def peak_sensitive_upheaval_index(components):
    max_component = max(components.values())
    avg_component = sum(components.values()) / len(components)

    # Weight toward the peak crisis, but include overall level
    return (max_component * 0.7) + (avg_component * 0.3)
```

Validation Framework

Historical Validation Tests

```
def validate_index_against_history(decades_data):
    results = []

    for decade, data in decades_data.items():
        score = calculate_composite_upheaval_index(data)
```

```
        results.append({
            'decade': decade,
            'composite_score': score['composite_score'],
            'expected_rank': get_historical_expectation(decade),
            'actual_rank': None  # To be calculated
        })

    # Rank decades by composite score
    results.sort(key=lambda x: x['composite_score'], reverse=True)
    for i, result in enumerate(results):
        result['actual_rank'] = i + 1

    return results

def get_historical_expectation(decade):
    # Expert/historical consensus on most turbulent decades
    rankings = {
        1960: 1,  # Assassinations, Vietnam, civil rights
        1970: 2,  # Watergate, oil crisis, Vietnam end
        2020: 3,  # COVID, Jan 6, polarization
        1940: 4,  # WWII
        2000: 5,  # 9/11, Iraq War
        1930: 6,  # Depression
        # ... etc
    }
    return rankings.get(decade, 10)
```

Sensitivity Analysis

```
def test_weight_sensitivity(decades_data):
    # Test different weighting schemes
    weight_schemes = [
        {'political_violence': 0.4, 'institutional_trust': 0.15, ...},  # Violence-heavy
        {'political_violence': 0.1, 'institutional_trust': 0.4, ...},   # Institution-heavy
        # Equal weights, etc.
```

```
]

correlations = []
for weights in weight_schemes:
    scores = [calculate_composite_upheaval_index(data, weights)
            for data in decades_data.values()]
    correlation_with_thrillers = calculate_correlation(scores, thriller_counts)
    correlations.append(correlation_with_thrillers)

return correlations
```

Expected Data Collection Needs

High-Priority Data (needed for all components)

1. **Political assassinations by decade**
2. **Major domestic riots/unrest events**
3. **Presidential/government scandals**
4. **War involvement (years, casualties, public support)**
5. **Recession data (duration, unemployment peaks)**
6. **Major terrorist attacks on US soil**

Medium-Priority Data (refinement)

1. **Income inequality statistics (Gini coefficient)**
2. **Political polarization measures**
3. **Supreme Court controversial decisions**
4. **Major corporate/financial scandals**

Future Enhancements

1. **Media trust polling data**
2. **Social cohesion surveys**

3. **Regional political differences**

This framework gives you a rigorous, defensible methodology while being flexible enough to adjust weights based on what correlates best with thriller production.