Quinal, Ana Rochelle Shillingford, Nigel BSCS – 3 Assignment1

Journal Summary

The study "Equilibrated Risk Set Alignment" by Yunfei Paul Li, Kathleen J. Propert, and Paul R. Rosenbaum focuses on a new matching method called optimal balanced risk set matching for non-experimental research. The method is applied to analyze the effects of cystoscopy and hydrodistension as a treatment for interstitial cystitis (IC), a chronic, nonlethal bladder disease. Since randomized trials are often impractical for studying treatments that are given in response to worsening symptoms, the study aims to improve the validity of observational comparisons by balancing covariates and minimizing differences between treated and untreated patients before treatment.

Steps to Implement Balanced Risk Set Matching

- Data Preparation: Load the dataset and preprocess it. Ensure that the data includes time-varying covariates and treatment assignment times.
- Risk Set Matching: For each treated patient at time t, find a control patient who has not been treated by time t and has a similar history of symptoms up to that point.
- Balancing Covariates: Use integer programming to ensure that the marginal distributions of symptoms are balanced in the matched treated and control groups.
- Optimization: Among all balanced matchings, select the one that minimizes the multivariate pretreatment covariate distance within matched pairs.
- Sensitivity Analysis: Perform sensitivity analysis to assess the robustness of the findings to hidden biases.

Risk Set Matching Algorithm

In observational studies, controlling for confounding variables is essential to draw valid causal inferences. Risk Set Matching is a statistical technique designed to mitigate confounding by pairing treated subjects with control subjects who are similar in key covariates and are at risk during the same time period. This method is particularly useful for time-to-event data, where the timing of events plays a critical role in the analysis.

Covariate Standardization: Standardized the covariates to have zero mean and unit variance using StandardScaler from scikit-learn.

Covariance Matrix Inversion: Calculated the inverse covariance matrix of the standardized covariates to compute Mahalanobis distances.

Distance Calculation: For each treated patient, computed the Mahalanobis distance to all potential control patients who were at risk at the same time or earlier.

Matching Criteria:

- Caliper: Set a caliper of 5.0 to define the maximum allowable distance for a valid match.
- Matching with Replacement: Allowed control patients to be matched multiple times to maximize the number of matched pairs.
- Pair Matching: Selected the control patient with the smallest Mahalanobis distance within the caliper for each treated patient.

Covariate Balance Assessment

We evaluated the effectiveness of the matching by calculating the standardized mean differences (SMD) for each covariate before and after matching:

Standardized Mean Difference:

$$\text{SMD} = \frac{\bar{X}_T - \bar{X}_C}{\sqrt{\frac{s_T^2 + s_C^2}{2}}}$$

An SMD less than 0.1 (in absolute value) typically indicates a negligible difference between groups.

Density Plots

Age Distribution

- Observation: The age distributions of the treated and control groups overlap almost perfectly after matching.

Blood Pressure Distribution

- Observation: Similar distributions indicate that blood pressure is well-balanced between the groups.

Glucose Distribution

- Observation: The glucose level distributions align closely, showing that the initial imbalance was corrected through matching.

Findings and Conclusions

The implementation of the Risk Set Matching algorithm successfully reduced covariate imbalances between the treated and control groups. Key findings include:

- Effective Covariate Balancing: The SMDs for age, blood pressure, and glucose were reduced to near zero after matching, indicating negligible differences between groups.
- High Matching Rate: By increasing the caliper and allowing matches with replacement, we matched nearly all treated patients, enhancing the representativeness of our sample.
- Improved Distribution Overlap: Density plots confirmed that the covariate distributions in the treated and control groups are virtually identical post-matching.

Conclusions:

- Validity of Causal Inferences: Achieving covariate balance is crucial for unbiased estimation of treatment effects. The improved balance suggests that any differences in outcomes between the groups are more likely attributable to the treatment rather than confounding variables.
- Utility of Risk Set Matching: This method proves effective for time-to-event data, especially when temporal factors are important. It allows for dynamic matching as patients become at risk or receive treatment over time.

Recommendations for Future Analysis:

- Outcome Evaluation: Analyze the outcomes of interest (e.g., survival time, event occurrence) using the matched dataset to estimate the treatment effect.
- Sensitivity Analysis: Conduct analyses to assess the robustness of results to potential unmeasured confounding or model assumptions.
- Alternative Matching Methods: Explore other matching techniques, such as propensity score matching or genetic matching, to compare results and validate findings.

Code and Graphs Integration

For a comprehensive understanding, the code used in this analysis and the generated graphs are essential. The code includes data generation, implementation of the RiskSetMatcher class, the matching process, covariate balance assessment, and visualization.

Data Generation

```
# Create sample data
np.random.seed(42)
n_patients = 1000

data = pd.DataFrame({
    'patient_id': range(n_patients),
    'time': np.random.uniform(0, 100, n_patients),
    'treatment': np.random.binomial(1, 0.3, n_patients),
    'age': np.random.normal(65, 10, n_patients),
    'blood_pressure': np.random.normal(120, 15, n_patients),
    'glucose': np.random.normal(100, 20, n_patients)
})
```

Running the Matcher

```
# Initialize and run the matcher
matcher = RiskSetMatcher(
    covariate_columns=['age', 'blood_pressure', 'glucose']
)

matched_pairs = matcher.match(data, caliper=5.0)
print(f"Number of matched pairs: {len(matched_pairs)}")
```

Assessing Balance

```
# Assess balance after matching
balance_stats_after = matcher.assess_balance(data, matched_pairs)
print(balance_stats_after)
```

Visualization

Love Plot

```
# Plot standardized mean differences
fig, ax = plt.subplots(figsize=(8, 6))

covariates = balance_pivot.index
before_smd = balance_pivot['Before Matching'].values
after_smd = balance_pivot['After Matching'].values

y_pos = np.arange(len(covariates))

ax.barh(y_pos + 0.2, before_smd, height=0.4, color='lightblue', label='Before Matching')
ax.barh(y_pos - 0.2, after_smd, height=0.4, color='steelblue', label='After Matching')

ax.set_yticks(y_pos)
ax.set_yticklabels(covariates)
ax.axvline(0, color='black', linewidth=0.8)
ax.set_xlabel('Standardized Mean Difference')
ax.set_title('Covariate Balance Before and After Matching')
ax.legend()
plt.tight_layout()
plt.show()
```

Density Plots

```
# Plotting distributions
for col in ['age', 'blood_pressure', 'glucose']:
    plt.figure(figsize=(8, 4))
    bins = np.linspace(
        min(matched_treated[col].min(), matched_control[col].min()),
        max(matched_treated[col].max(), matched_control[col].max()),
        30
    )

    plt.hist(
        matched_treated[col],
        bins=bins,
        alpha=0.6,
        label='Treated',
        density=True,
```

```
        color='blue'
    )

    plt.hist(
        matched_control[col],
        bins=bins,
        alpha=0.6,
        label='Control',
        density=True,
        color='orange'
    )

    plt.title(f'Distribution of {col.capitalize()} After Matching')
    plt.xlabel(col.capitalize())
    plt.ylabel('Density')
    plt.legend()
    plt.tight_layout()
    plt.show()
```