Record Linkage and Entity Resolution

Jared Edgerton

Why Record Linkage Matters

Many social datasets:

- Refer to the same entities
- Use inconsistent identifiers
- Were never designed to be merged

Linkage constructs the unit of analysis.

Entity Resolution as Measurement

Entity resolution determines:

- Who exists
- Which records correspond
- How units persist over time

This is a measurement problem, not clerical cleanup.

Deterministic vs Probabilistic Logic

```
# Deterministic rule
if name_match and dob_match:
    link = True
else:
    link = False
```

```
# Probabilistic logic
p_link = f(name_similarity, dob_similarity, location_similarity)
link = (p_link > threshold)
```

Thresholds encode tradeoffs.

Blocking Strategy (Pseudocode)

```
# Input: records A, records B
blocks = {}

for record in A:
    key = blocking_key(record)      # e.g., first letter + year
    blocks[key].append(record)

candidate_pairs = []
for key in blocks:
    candidate_pairs.extend(cartesian(blocks[key], B[key]))
```

Blocking reduces complexity but risks missed matches.

Feature Construction for Matching

```
# Construct comparison features
features = {
    'name_sim': string_similarity(a.name, b.name),
    'dob_sim': date_similarity(a.dob, b.dob),
    'loc_sim': geo_distance(a.loc, b.loc)
}
```

Distances are evidence, not decisions.

Fellegi–Sunter Logic (Conceptual)

```
# Likelihood ratio for match vs non-match
LR = P(features | match) / P(features | non_match)

if LR > cutoff:
    classify as match
elif LR < lower_cutoff:
    classify as non_match
else:
    send to clerical review
```

Uncertainty is explicit.

EM-Style Estimation (Conceptual)

```
# Initialize parameters
theta = initialize()

repeat until convergence:
    # E-step: estimate match probabilities
    for pair in candidate_pairs:
        pair.p_match = compute_p(pair, theta)

    # M-step: update parameters
    theta = update_parameters(candidate_pairs)
```

Linkage models are estimated, not assumed.

Many-to-One Resolution

```
# Resolve conflicts
for entity in entities:
    candidates = linked_records(entity)
    keep = argmax(candidates, key = p_match)
    discard others
```

Resolution rules affect downstream counts.

Evaluation with Labeled Data

```
precision = true_matches / predicted_matches
recall    = true_matches / actual_matches
```

No single metric dominates.

Sensitivity Analysis

```python
for threshold in thresholds:
    results = run_linkage(threshold)
    compare_outcomes(results)
```

Robustness matters more than point estimates.

Error Propagation

```
# Downstream model
Y = model(linked_data)

# Linkage uncertainty propagates into Y
Y_variance += linkage_uncertainty
```

Ignoring linkage error biases inference.

Documentation Requirements

Every linkage pipeline should record:

- Blocking rules
- Similarity metrics
- Thresholds
- Evaluation results
- Known failure modes

Undocumented linkage is irreproducible.

What We Emphasize in Practice

- Treat linkage as measurement
- Preserve uncertainty
- Evaluate tradeoffs explicitly
- Document all design choices

Discussion

- Where are false positives most costly?
- When are false negatives more dangerous?
- How should linkage uncertainty be reported?