CSCE 670 - Information Storage and Retrieval

Lecture 8: Evaluation (and Quiz 1)

Yu Zhang

yuzhang@tamu.edu

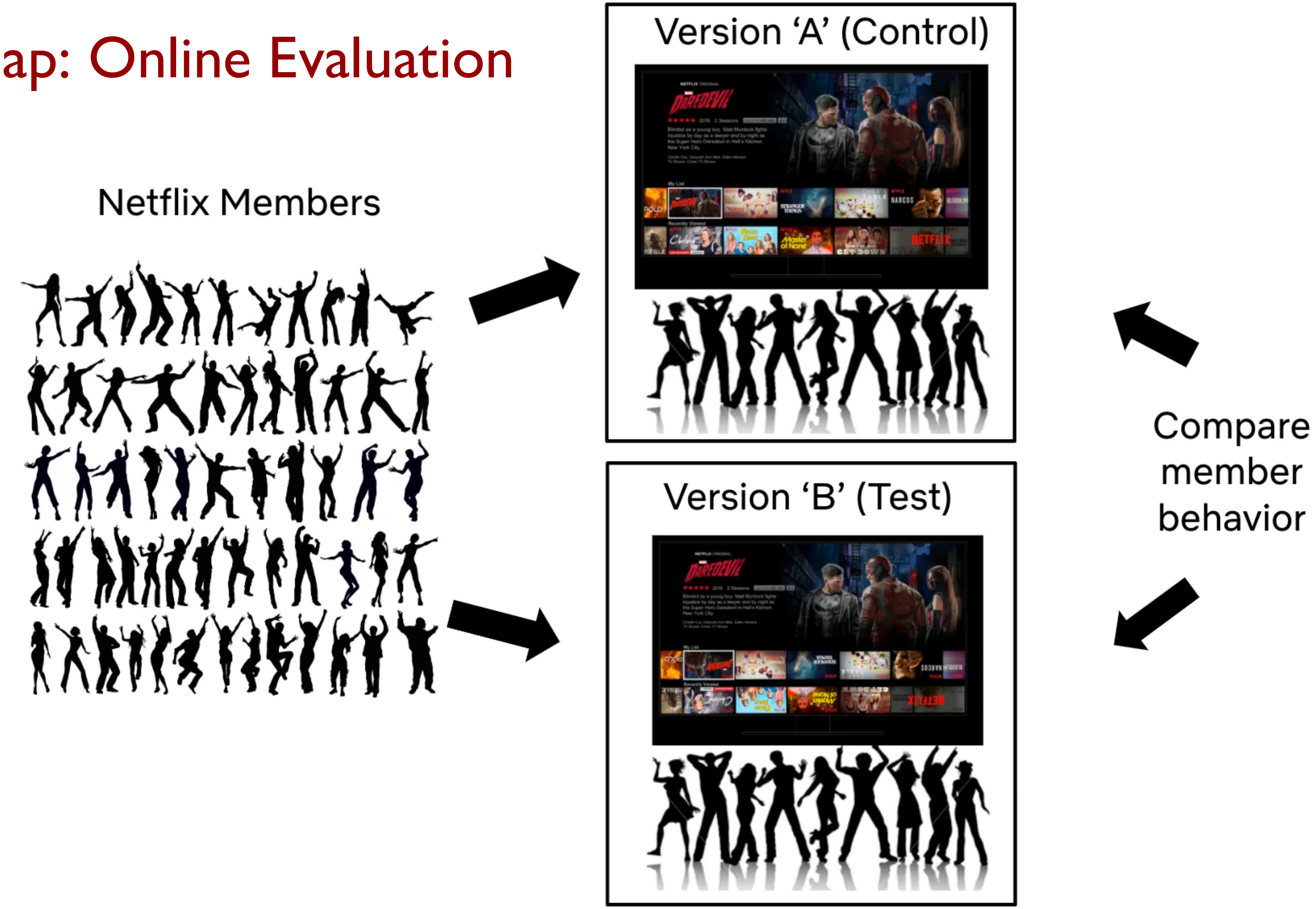
September 18, 2025

Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html

Recap: Offline Evaluation

- Hypothesis: A new search engine (e.g., based on *SuperRank*) is better than an old one (e.g., based on BM25)
- What we need:
  - Documents (representative of our collection),
  - Queries (that we hope are representative of what our users will ask), and
  - Relevance judgments (can be expensive to collect and noisy)
- Metrics:
  - Precision, Recall, F1
  - P@*k*, MAP, NDCG@*k*

Recap: Online Evaluation



True Merit vs. Randomness

- Can we conclude from this offline test that Algorithm *B* outperforms Algorithm *A*?

	NDCG@5
Algorithm *A*	0.7000
Algorithm *B*	0.7001

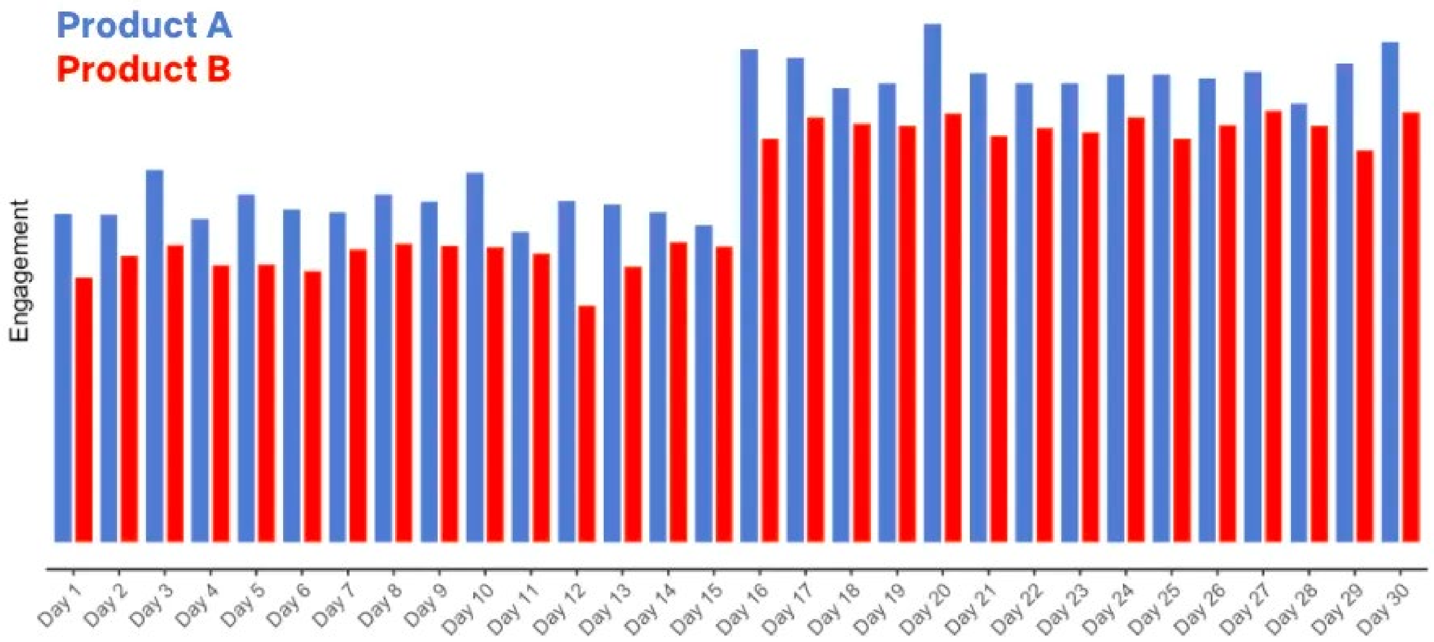
- Can we conclude from this online test that Algorithm *B* outperforms Algorithm *A*?

	User Click-Through Rate
Algorithm *A*	0.3000
Algorithm *B*	0.3100

- We need statistical significance tests!

Statistical Significance Tests for Evaluating a Search Engine

- Step 1: Evaluate Algorithms *A* and *B* under different experimental conditions
  - Query types (offline)
  - Time of experiment (online)
  - Random seeds (if the algorithm involves randomness)
  - …



Statistical Significance Tests for Evaluating a Search Engine

- Step 2: Compare the metrics of Algorithms *A* and *B* and examine whether they are likely drawn from different probability distributions

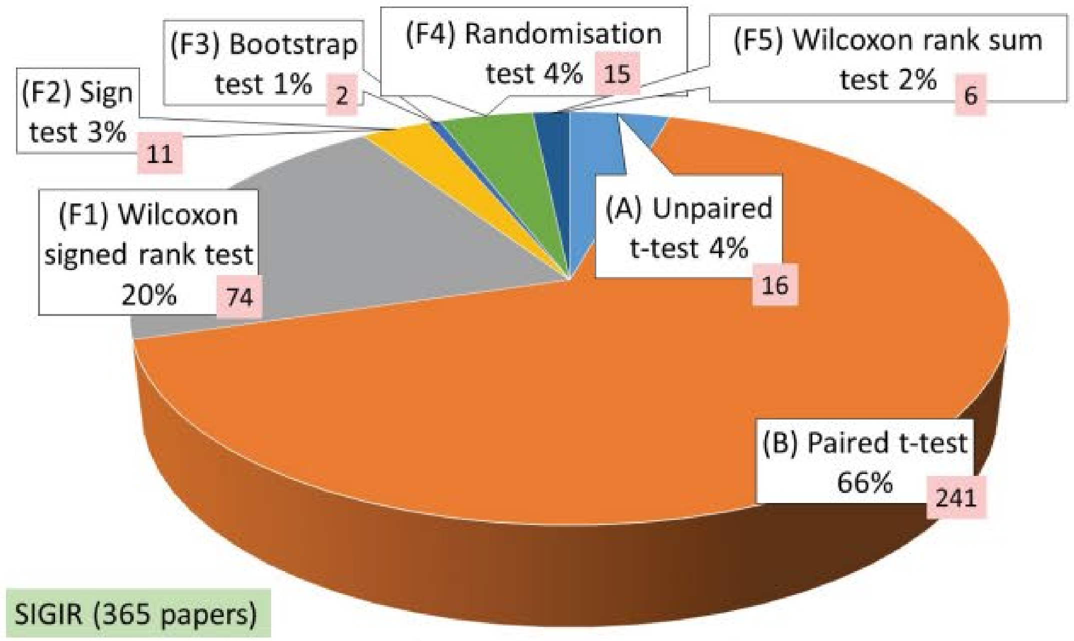
	Algorithm *A*	Algorithm *B*	Difference
Condition 1	x_1	y_1	$y_1 - x_1$
Condition 2	x_2	y_2	$y_2 - x_2$
…	…	…	…
Condition *N*	x_N	y_N	$y_N - x_N$

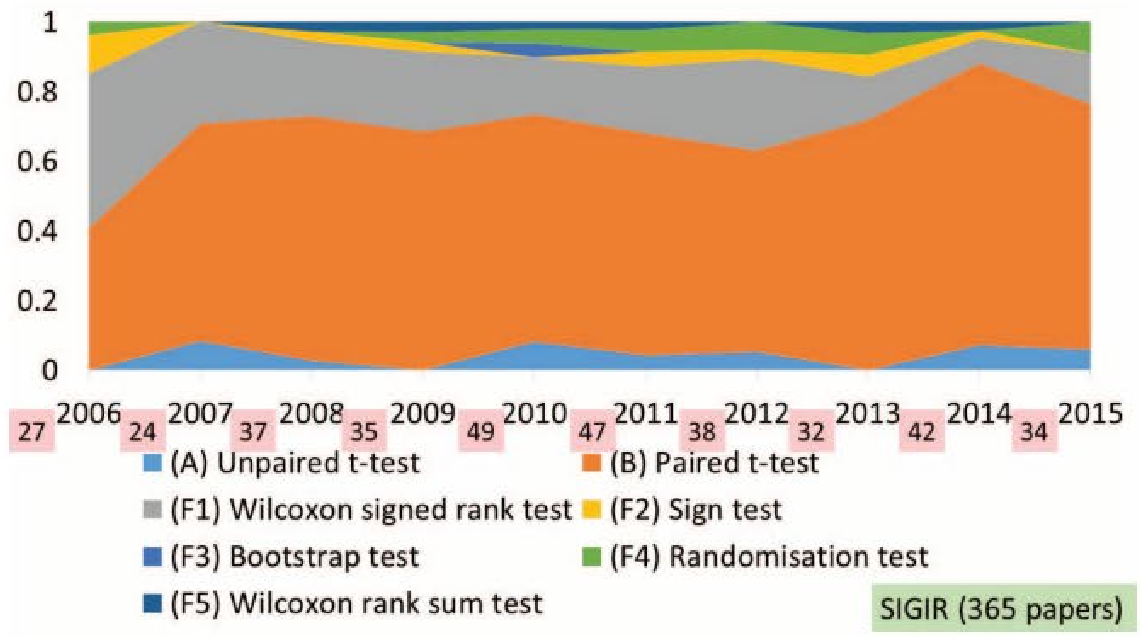
- Null Hypothesis: The case you hope to rule out
  - $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ are drawn from two distributions with the same mean, OR
  - $\{y_i - x_i\}_{i=1}^N$ are drawn from a distribution with mean 0

- Statistical Significance Test: Using probability theory to show that the likelihood of the null hypothesis being true is very small (e.g., < 0.01).

Statistical Significance Tests for Evaluating a Search Engine

- *Statistical Significance, Power, and Sample Sizes: A Systematic Review of SIGIR and TOIS, 2006-2015.* SIGIR 2016.
  - The most commonly used tests in IR: Paired t-test (66%), Wilcoxon signed rank test (20%), and Unpaired t-test (4%)





Paired t-test

- Null Hypothesis: $\{y_i - x_i\}_{i=1}^{N}$ are drawn from a distribution with mean 0
- Step 1: Calculate the t-statistic

$$t = \frac{\text{mean of } \{y_i - x_i\}_{i=1}^{N}}{\left(\text{standard deviation of } \{y_i - x_i\}_{i=1}^{N}\right)/\sqrt{N}}$$

- Step 2: Calculate the "*degrees of freedom*": $N - 1$
- Step 3: Look up the t-statistic in a t-distribution table (you need to know $N - 1$ to find the correct row) to obtain the p-value
  - p-value = Prob[the difference between Algorithms *A* and *B* is due to randomness]
  - If p-value < 0.05, then Prob[the difference between Algorithms *A* and *B* is due to true merit] > 0.95, and we say the difference is statistically significant.

Paired t-test

- We can also do this in Python:

python Copy Edit

```python
from scipy.stats import ttest_rel

# Sample data
X = [0.5, 0.4, 0.6, 0.3, 0.2, 0.4, 0.5, 0.3, 0.2, 0.5]
Y = [0.3, 0.2, 0.5, 0.2, 0.1, 0.3, 0.4, 0.2, 0.1, 0.4]

# Calculate t-statistic and p-value
t, p = ttest_rel(X, Y)

# Print p-value
print(p)
```

- In this example, p-value = 8.538e-06

Wilcoxon Signed Rank Test

- Paired t-test assumes that $\{y_i - x_i\}_{i=1}^N$ are drawn from a normal distribution
- Wilcoxon signed rank test has a much weaker assumption: $\{y_i - x_i\}_{i=1}^N$ are drawn from a symmetric distribution around the mean

- Null Hypothesis: $\{y_i - x_i\}_{i=1}^N$ are drawn from a distribution with mean 0

- Example: $\{y_i - x_i\}_{i=1}^N$ = {0.20, -0.10, 0.30, -0.05}

- Step 1: Compute $|y_i - x_i|$
  - 0.20, 0.10, 0.30, 0.05

- Step 2: Sort these values and assign ranks
  - 0.05 (rank=1), 0.10 (rank=2), 0.20 (rank=3), 0.30 (rank=4)

Wilcoxon Signed Rank Test

- Null Hypothesis: $\{y_i - x_i\}_{i=1}^{N}$ are drawn from a distribution with mean 0
- Example: $\{y_i - x_i\}_{i=1}^{N}$ = {0.20, -0.10, 0.30, -0.05}
- Step 1: Compute $|y_i - x_i|$
  - 0.20, 0.10, 0.30, 0.05
- Step 2: Sort these values and assign ranks
  - 0.05 (rank=1), 0.10 (rank=2), 0.20 (rank=3), 0.30 (rank=4)
- Step 3: Calculate the signed-rank sum
  - T = (-1) + (-2) + (+3) + (+4) = 4
  - Intuition: If the Null Hypothesis holds, T should be close to 0.
- Step 4: Look up T in the table to obtain the p-value

Wilcoxon Signed Rank Test

- We can also do this in Python:

```python
from scipy.stats import wilcoxon

# Sample data
X = [0.5, 0.4, 0.6, 0.3, 0.2, 0.4, 0.5, 0.3, 0.2, 0.5]
Y = [0.3, 0.2, 0.5, 0.2, 0.1, 0.3, 0.4, 0.2, 0.1, 0.4]

# Perform Wilcoxon Signed-Rank Test
stat, p = wilcoxon(X, Y)

# Print p-value
print(p)
```

- In this example, p-value = 0.00195

Unpaired t-test

- What if a paired comparison is NOT feasible?
  - E.g., when the two IR models use entirely different architectures with different hyperparameter settings, and we are conducting an offline evaluation

- Null Hypothesis: $\{x_i\}_{i=1}^{M}$ and $\{y_j\}_{j=1}^{N}$ are drawn from two distributions with the same mean

- If we can assume the two distributions have the same variance:

$$t = \frac{\bar{x} - \bar{y}}{s_p\sqrt{\frac{1}{M} + \frac{1}{N}}}, \qquad \text{where } \bar{x} = \frac{x_1 + \cdots + x_M}{M}, \qquad \bar{y} = \frac{y_1 + \cdots + y_N}{N}$$

$$\text{and } s_p = \sqrt{\frac{(M-1)s_X^2 + (N-1)s_Y^2}{M+N-2}}$$

Unpaired t-test

- If we cannot assume the two distributions have the same variance (Welch's t-test):

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_X^2}{M} + \frac{s_Y^2}{N}}}, \qquad \text{where } \bar{x} = \frac{x_1 + \cdots + x_M}{M}, \qquad \bar{y} = \frac{y_1 + \cdots + y_N}{N}$$

python Copy Edit

```python
from scipy.stats import ttest_ind

# Sample data
X = [0.5, 0.4, 0.6, 0.3, 0.2, 0.4, 0.5, 0.3, 0.2, 0.5]
Y = [0.3, 0.2, 0.5, 0.2, 0.1, 0.3]  # 4 elements removed

# Unpaired t-test (equal variance)
t_equal, p_equal = ttest_ind(X, Y, equal_var=True)

# Welch’s t-test (unequal variance)
t_unequal, p_unequal = ttest_ind(X, Y, equal_var=False)
```

Quiz 1

Thank You!

Course Website: https://yuzhang-teaching.github.io/CSCE670-F25.html