Continual Subjective Evaluation Method of Speech by Merging Sort-based Preference Tests Towards Ever-Expanding Corpus of Human Ratings

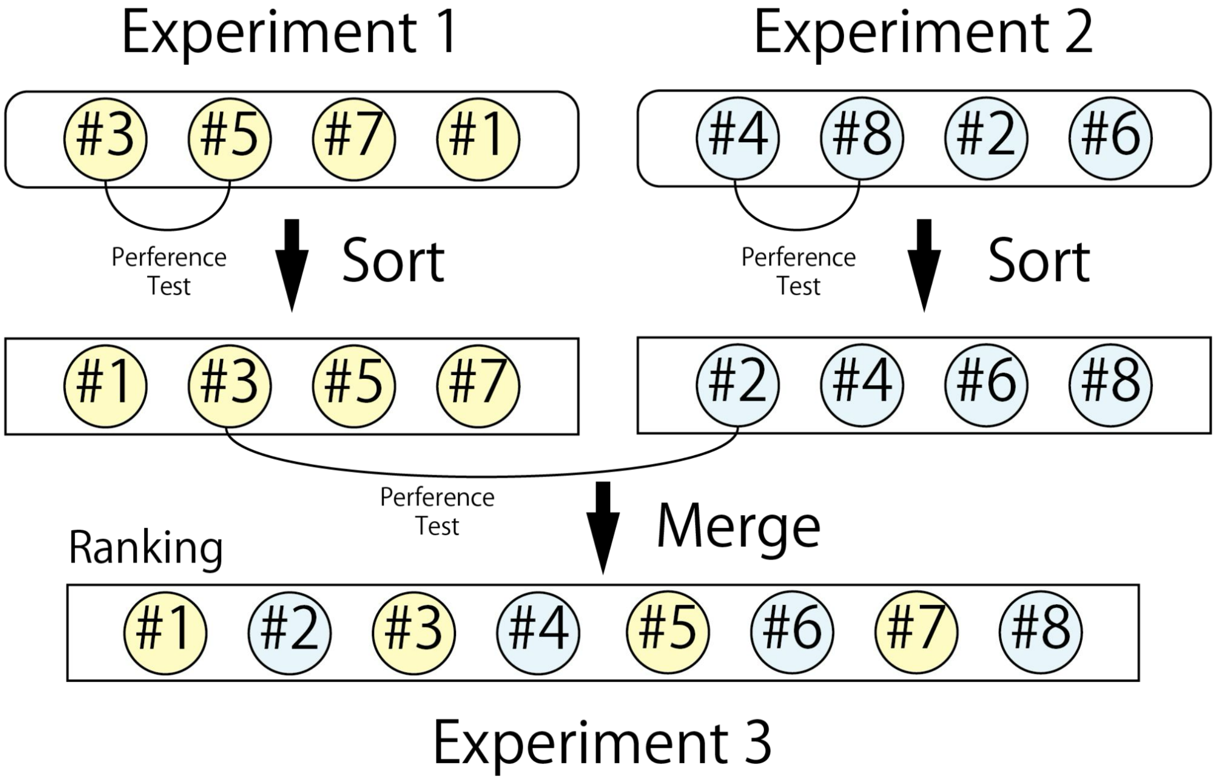
Yusuke Yasuda* Junichi Yamagishi*, and Tomoki Toda**
*National Institute of Informatics, **Nagoya University

Motivation

- Large-scale subjective rating corpora emerged.
  - Targeting to training corpus for automatic quality prediction.
- Current limitation of corpus construction for subjective ratings:
  - High cost and limited size
  - Scores are context-dependent
  - Requirement to single-shot experiment
- How can we enlarge subjective corpus step-by-step?
  - →Continual subjective evaluation

Continual Subjective Evaluation: Task Definition

- Rank systems by solving a loop of two subproblems:
- (1) sorting subsets of systems in the quality order
- (2) merging the subsets of sorted systems into a single ranking.



Continual Subjective Evaluation: Challenges

- (1) Divisibility: Evaluations must be divided into several experiments to add systems at different time points;
- (2) Consistency: The derived ranking from multiple evaluations should be consistent;
- (3) Cost-efficiency: Cost efficiency is required for evaluations to be continual up to a large-scale system set.

Continual Subjective Evaluation: Limitations of existing methods

- MOS:
  - ❌Scores are not consistent across different experiments evaluating different system sets.
  - →(1) Single-shot requirement: experiments can not be divided or merged.
  - →(2) Ranking consistency is not expected.
  - ✅Cost efficient.
- Preference:
  - ❌Cost inefficient due to the huge number of pair combinations.
  - →(3) Not scalable to a large number of systems.
    - Normally, about 5 pairs are evaluated.
  - ✅Relative scores.
  - →(1), (2) Can derive a consistent ranking even if evaluation is divided into several experiments.

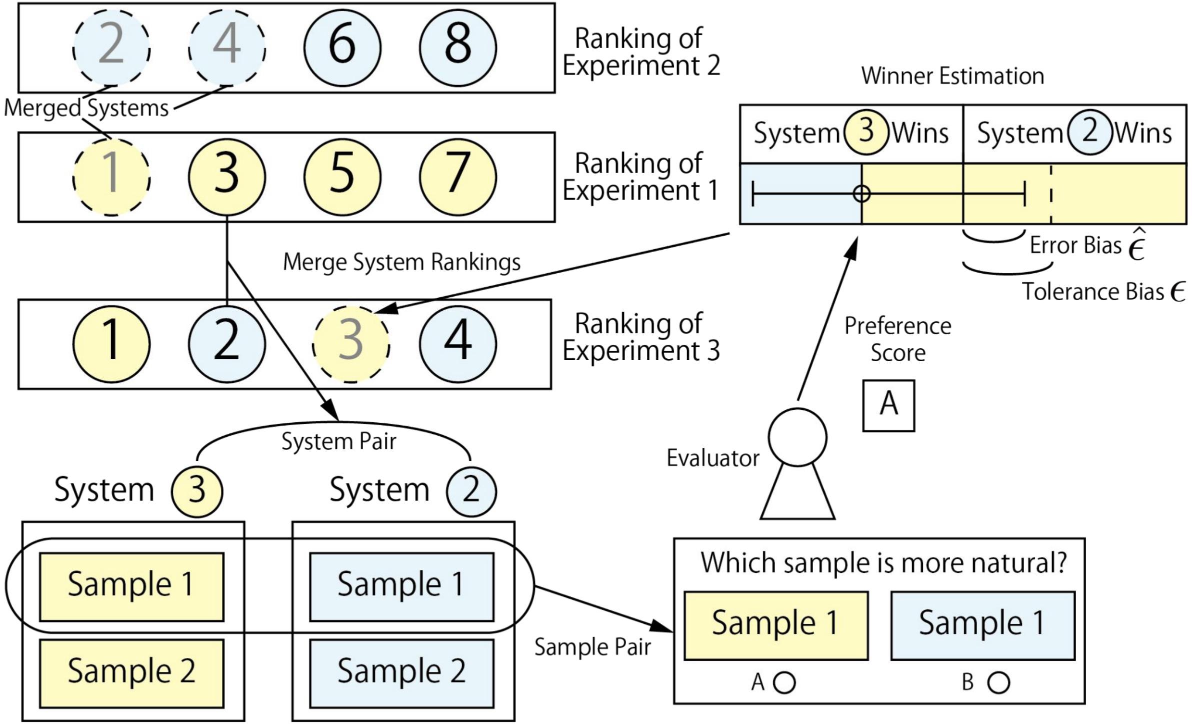
Contributions

- We define the continual subjective evaluation as a new subjective evaluation task that can expand systems to evaluate over time;
- We propose a method to realize the continual subjective evaluation based on preference tests and merge- and sort-based online learning;
- We conduct an iteration of the continual subjective evaluation in three experiments to derive a ranking of 60 systems;
- Our experiments show that our method can realize the continual subjective evaluation by deriving a ranking of 60 systems efficiently from preference tests evaluating 216 pairs.

Proposed Method

Proposed Method: Preference Evaluation with Online Learning

- Sorting and merging algorithm are integrated with listening test system.
- Pairs are selected based on the algorithms.
- Minimum evaluations to rank are allocated.



Algorithm for merging: MERGE

- Based on merge algorithm but stochastic.
- Merging two sorted sets S1 and S2.
- O(|S1| + |S2|) pair complexity to rank.

Algorithm MERGE

Input: Sorted sets S_1, S_2, bias ϵ, confidence δ.
Initialize: $i = 1, j = 1$ and $O = \emptyset$.

> **while** $i \leq |S_1|$ and $j \leq |S_2|$ **do**
>> **if** $S_1(i) = \text{COMPARE}(S_1(i), S_2(j), \epsilon, \delta)$ **then**
>>> append $S_2(j)$ at the end of O and $j = j + 1$.
>>
>> **else**
>>> append $S_1(i)$ at the end of O and $i = i + 1$.
>>
>> **if** $i \leq |S_1|$ **then**
>>> append $S_1(i : |S_1|)$ at the end of O.
>>
>> **if** $j \leq |S_2|$ **then**
>>> append $S_2(j : |S_2|)$ at the end of O.

Output: Sorted set O

Algorithm for sorting (1): MERGE-RANK

- Based on merge-sort algorithm but stochastic version.
- Divide and conquer approach.
- O(|S|log|S|) pair complexity to sort.

Algorithm MERGE-RANK

Input: Set S, bias ϵ, confidence δ.

$$S_1 = \text{MERGE-RANK}(S(1 : \lfloor |S|/2 \rfloor), \epsilon, \delta)$$
$$S_2 = \text{MERGE-RANK}(S(\lfloor |S|/2 \rfloor + 1 : |S|), \epsilon, \delta)$$

Output: $\text{MERGE}(S_1, S_2)$

Algorithm for sorting (2): INSERT-RANK

- Based on insert-sort algorithm but stochastic version.
- Incremental approach.
- O(|S|) pair complexity to sort at the best case.
- O(|S|^2) pair complexity to sort at the worst case.

Algorithm INSERT-RANK

Input: Set S, bias ϵ, confidence δ.
Initialize: $i = 1, j = 2$.

for $j = 2, \ldots, |S|$ **do**
  $i = j - 1$
  while $i > 0$ AND COMPARE$(S(i), S(j), \epsilon, \delta) = S(i)$ **do**
    Insert in place $S(i+1) \leftarrow S(i)$
    $i = i - 1$
  Insert in place $S(i+1) \leftarrow S(j)$

Output: Sorted set S

Algorithm for winner determination: COMPARE

- Listener preferences are stochastic.
- Sorting and merging algorithm need to know a winner of a pair to rank.
- COMPARE algorithm determines a winner from preferences with at most error bias ε and error probability δ.

Algorithm COMPARE

Input: element pair i, j, bias ϵ, confidence δ.

Initialize: $\hat{p}_{ij} = \frac{1}{2}, m = \frac{1}{2\epsilon^2} \log \frac{2}{\delta}, r_{ij} = 0, w_{ij} = 0$.

Define: $\hat{c}(r) = \sqrt{\frac{1}{2r} \log \frac{4r^2}{\delta}}$ if $r > 0$ else $\frac{1}{2}$.

Define: $\hat{\epsilon}(r, \hat{p}) = \hat{c}(r) - |\hat{p} - \frac{1}{2}|$.

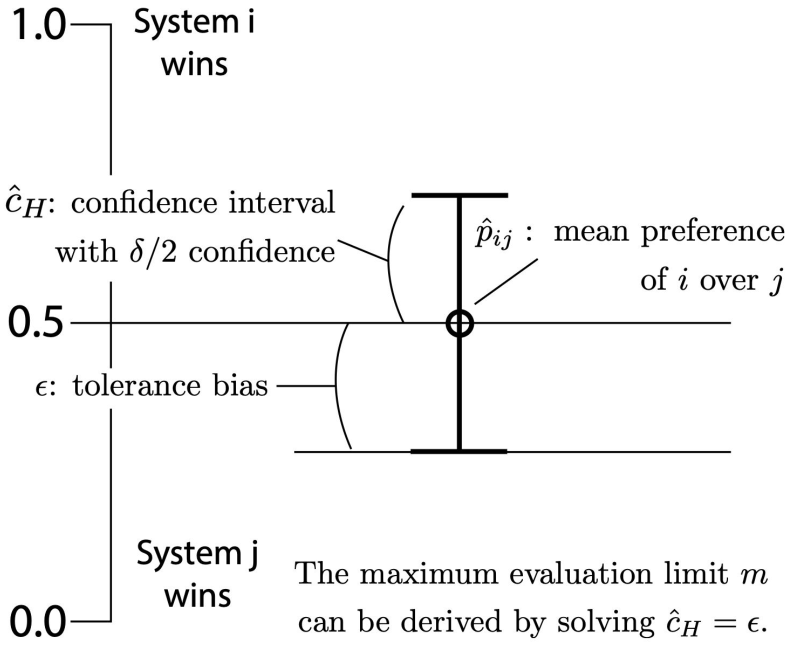
  while $\epsilon \leq \hat{\epsilon}(r_{ij}, \hat{p}_{ij})$ and $r_{ij} \leq m$ **do**

    Compare i and j. **if** i wins, $v_{ij} = 1$ **else** $v_{ij} = 0$.

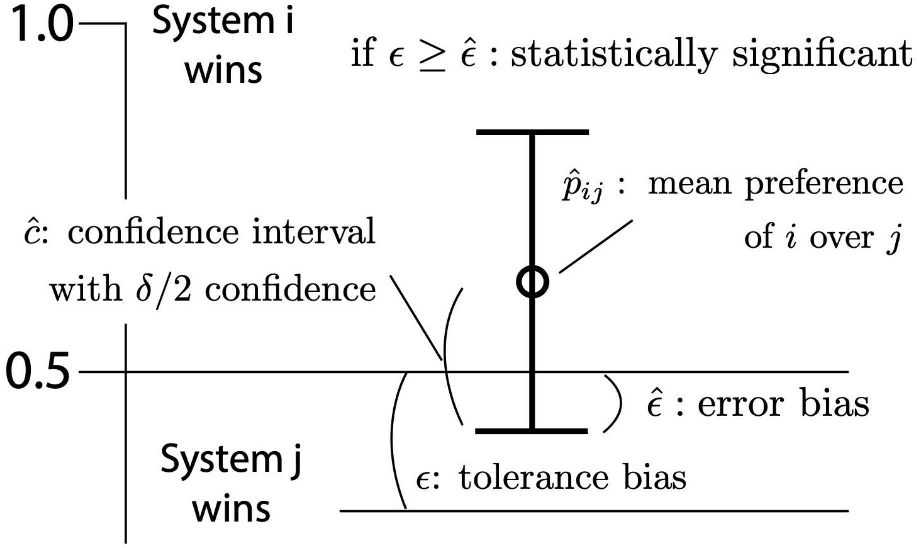
    $w_{ij} = w_{ij} + v_{ij}, r_{ij} = r_{ij} + 1, \hat{p}_{ij} = \frac{w_{ij}}{r_{ij}}$.

if $\hat{p}_{ij} \leq \frac{1}{2}$ **Output:** winner j **else Output:** winner i

Algorithm for winner determination: COMPARE



The worst case



The best case

Experimental Evaluation

Experimental settings

- Dataset: BVCC (VoiceMOS Challenge 2022)
- TTS systems in Blizzard Challenge, Voice Conversion Challenge, and more.
- Top 60 systems are selected.
- Divided into two subsets: odd and even rank set bases on MOS ranking
- Three experiments are conducted.
- Experiment 1: sorting the set 1 (30 systems)
- Experiment 2: sorting the set 2 (30 systems) and merging set 1 and 2 partially (10 systems)
- Experiment 3: merging the rest of set 1 and 2 (50 systems)
- Speech samples were evaluated on naturalness via crowdsourcing.

Experiment No.	1	2	3
Sort Algorithm	Insert Rank	Merge Rank	-
Merge Algorithm	-	Merge	Merge
#Sort Systems	30	30	-
#Merge Systems	-	10	50
#Scores in Budget	24,960	24,960	15,540
#Convergence Cost	14,977	19,658	9,761
#Evaluated Pairs	70	98	48
#Significant Pairs	28	49	21
#Max Cost per Pair	528	413	465
#Min Cost per Pair	219	60	127

Table 1: *Settings and results of three experiments.*

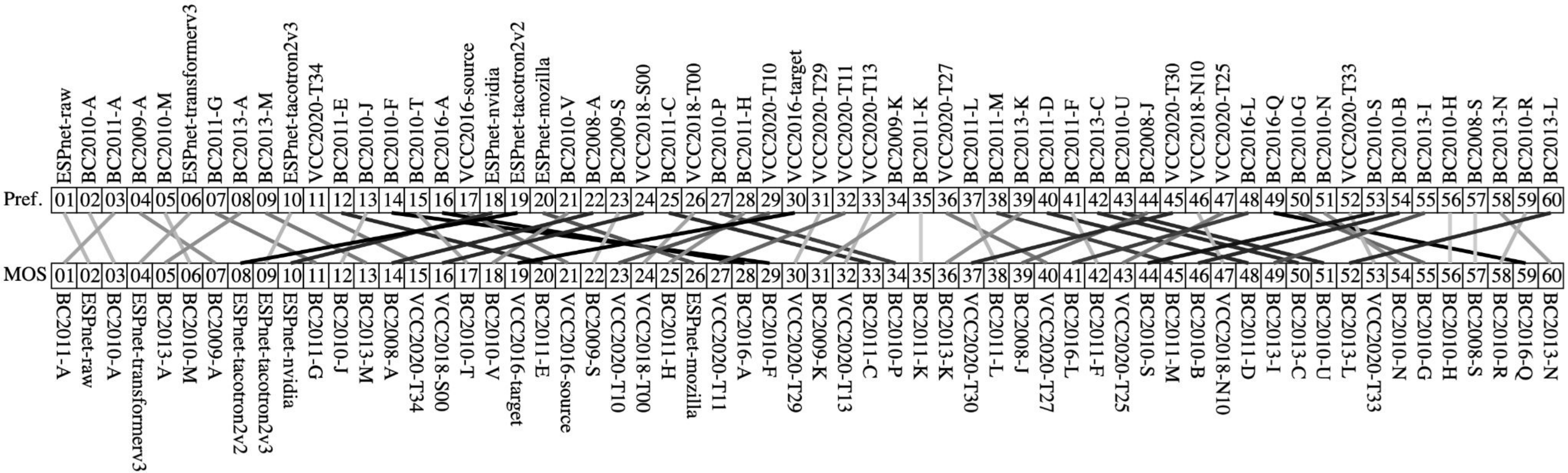
Results: Overview

- 60 systems were ranked with our method.
  - The continual subjective evaluation was feasible.
- Other findings:
  - Sorting and merging can be seamlessly evaluated with MERGE-RANK and MERGE.
  - INSERT-RANK was more efficient than MERGE-RANK.
  - INSERT-RANK was less performant than MERGE-RANK for crowdsourcing.

Experiment No.	1	2	3
Sort Algorithm	Insert Rank	Merge Rank	-
Merge Algorithm	-	Merge	Merge
#Sort Systems	30	30	-
#Merge Systems	-	10	50
#Scores in Budget	24,960	24,960	15,540
#Convergence Cost	14,977	19,658	9,761
#Evaluated Pairs	70	98	48
#Significant Pairs	28	49	21
#Max Cost per Pair	528	413	465
#Min Cost per Pair	219	60	127

Table 1: *Settings and results of three experiments.*

Results: Ranking

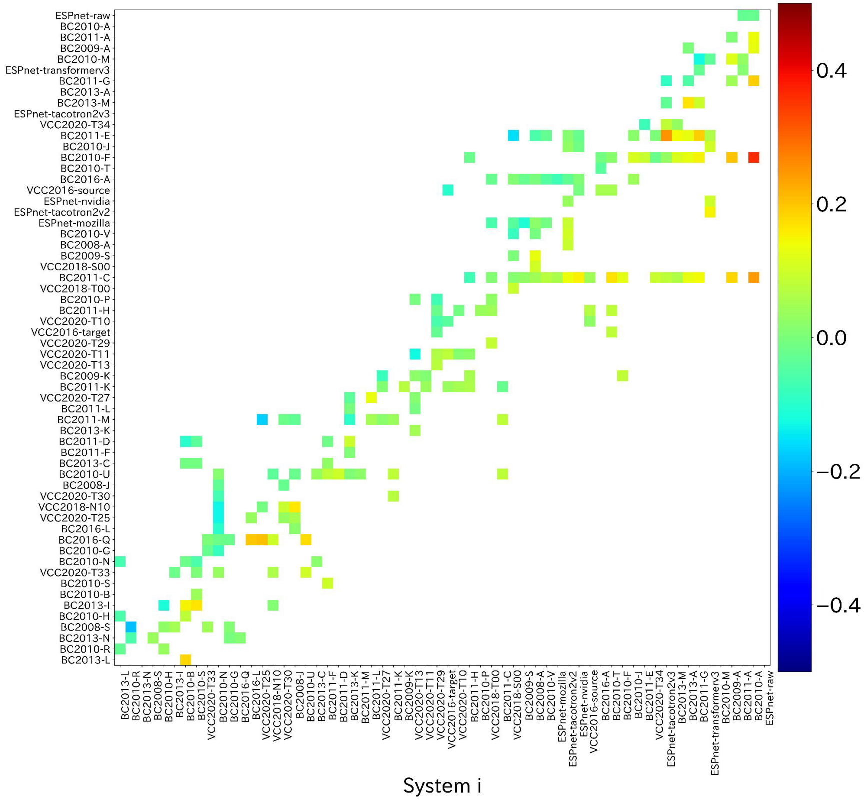


- We obtained similar ranking to MOS:
- Kendall’s tau: 0.798
- Spearman correlation coefficient: 0.943

- However, our ranking was not exactly same as MOS.
- Possible reasons:
- Lack of statistical differences in many pairs in BVCC corpus.
- Contraction bias in MOS.
- Our method can be used for detail evaluation.

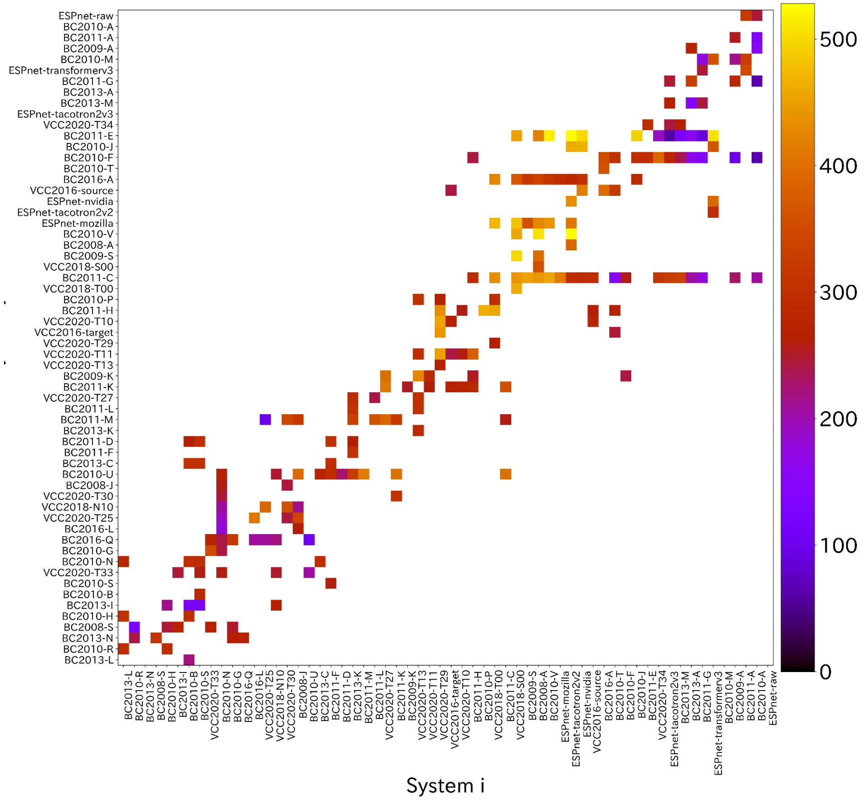
Results: Preference distributions

- Pairs with similar quality were selectively evaluated.
- Diagonal region: pairs with similar scores.
- Off-diagonal region: pairs with different scores.



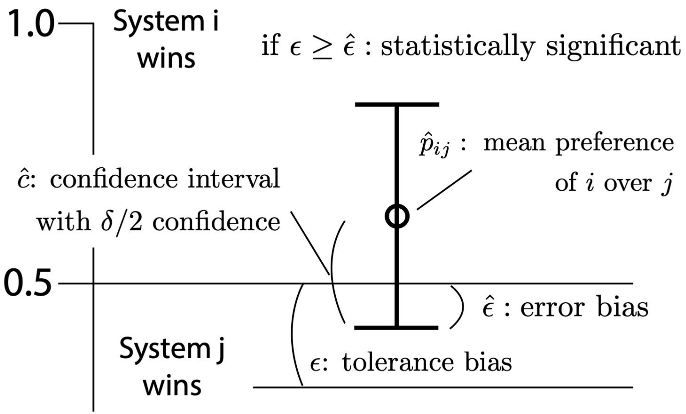
Results: Evaluation cost distribution

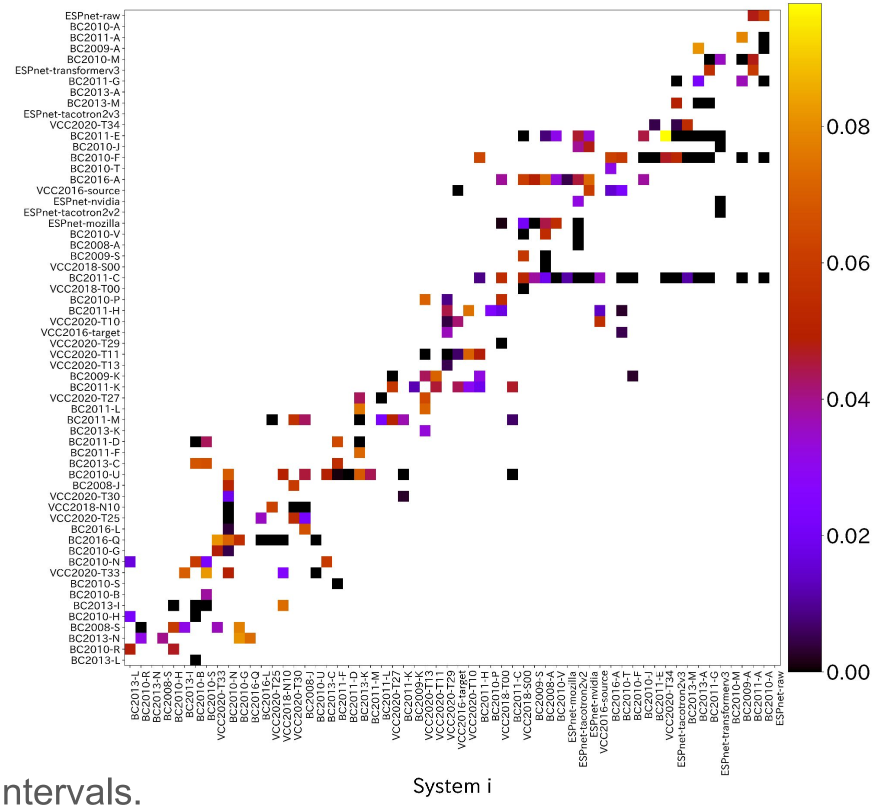
- Pairs with similar scores were evaluated many times.
- Pairs with different scores were evaluated few times.



Results: Evaluation error distribution.

- All pairs achieved errors below the threshold.
- Many pairs had near zero evaluation errors.





Definition of evaluation error: overlap of confidence intervals.

Conclusion

- This study defined a continual subjective evaluation of speech to keep expanding systems and scores in a subjective evaluation corpus
- We proposed our method to realize the continual subjective evaluation based on preference-based online learning.
- Future works:
  - Application to the automatic quality evaluation.
  - Application to other media evaluation than speech.