Synthesis of Partial Rankings of Points of Interest Using Crowdsourcing

Ilkcan Keles

Simonas Saltenis

Christian S. Jensen

Outline

- Introduction
- Problem Definition
- Related Work
  - Crowdsourcing
  - Rank Aggregation
- PointRank
  - Preliminaries
  - Overview
  - Determining the Next Question
  - Processing the Question
- Experimental Results
- Conclusion

Introduction

- Top-k Spatial Keyword Queries
  - Given
    - User Location – latitude, longitude pair
    - Keywords
    - *k* – the size of the result list
  - Return
    - A ranked list of *k* PoIs wrt a ranking function
- Motivation
  - No existing work on the quality of the ranking functions
  - No way to compare the ranking functions
  - No mathematical definition of the best ranking
  - Best ranking depends on preference of the users
  - A methodology to construct the best ranking is needed

Problem Definition

- *D* – the set of PoIs returned in response to a spatial keyword query.
- Pairwise relevance relation $\prec$ on D
  - Irreflexive
  - Transitive
  - Asymmetric
- Each element $p_i \prec p_j$ is called as pairwise relevance.
- If for each pair (p_i, p_j), $p_i \prec p_j$ or $p_j \prec p_i$ then it is a total order.
- The problem is to design a model
  - To construct a pairwise relevance relation $\prec$ on *D* via crowdsourcing.
- The synthesized relation $\prec$ should be
  - Similar to a ground-truth relation
  - Synthesized in an efficient manner.

PointRank / Preliminaries

- Pairwise relevance question
  - A pair of PoIs
  - Asks which of the two PoIs are relevant to the query
- Assignment
  - Assignment of a pairwise relevance question to a worker
- Three possible answers
  - First PoI is more relevant
  - Second PoI is more relevant
  - They are incomparable
- Consensus
  - If there is not a significant change in the answers in two iterations
- Chi-square test is used to check the significance of the change
  - P-value is the probability that the change is due to chance

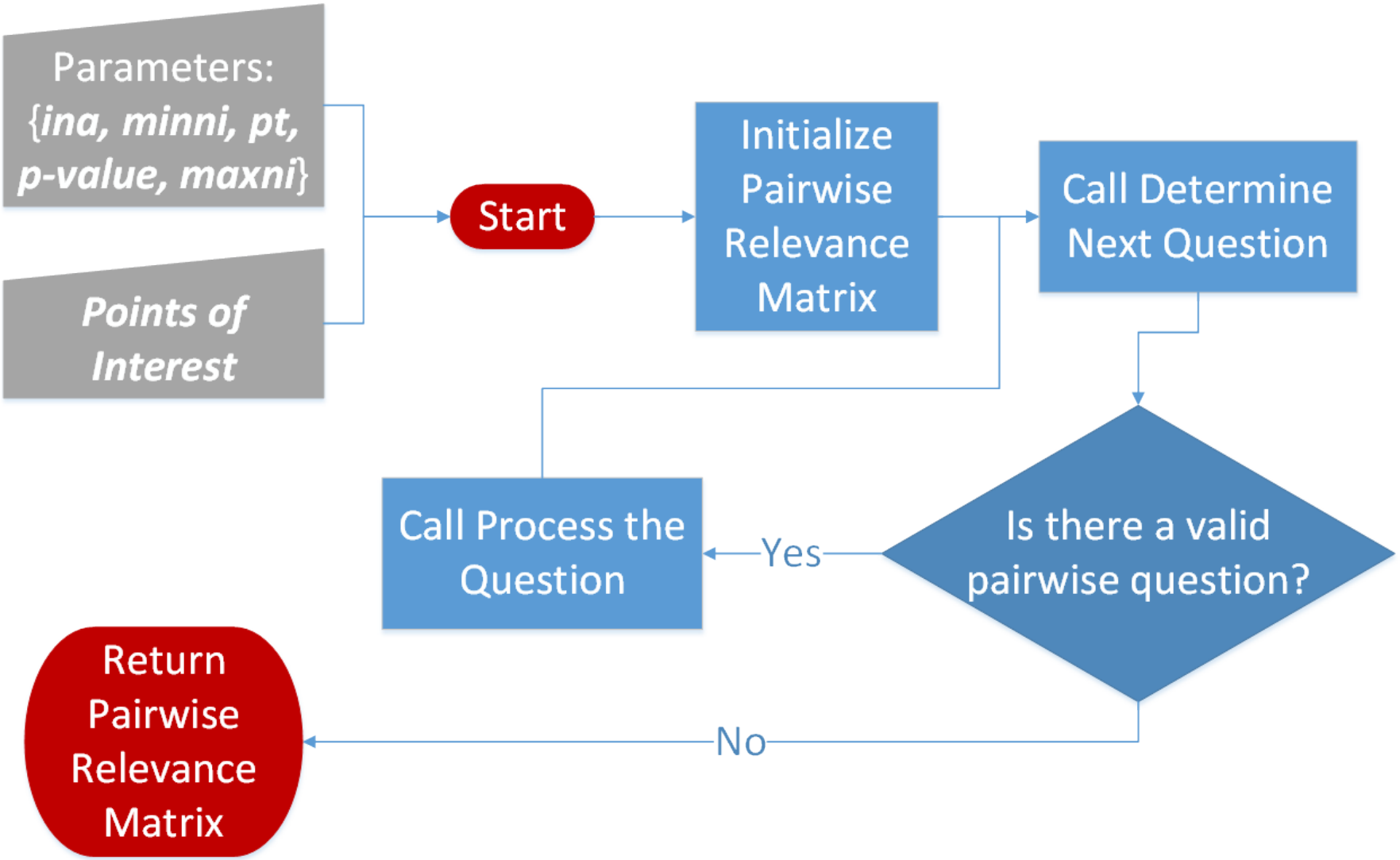
PointRank / Overview

- Edge weighted directed graph
  - To store the answers of the workers
- Pairwise Relevance Matrix (PRM)
  - To store the output of the algorithm (pairwise relevance relation)
- A cell in PRM *M* can have one of the possible values:
  - *M*[i,j] = 1 encoding $p_i \prec p_j$
  - *M*[i,j] = 0 encoding that p_i and p_j are incomparable
  - *M*[i,j] = -1 encoding $p_j \prec p_i$
  - *M*[i,j] = 2 encoding that (p_i, p_j) is not processed.
  - *M*[i,j] = 3 encoding that the algorithm cannot decide about (p_i, p_j)
- *M* has the following properties:
  - Transitivity – If M[i,j] = 1 and M[j,k] = 1 then M[i,k] = 1.
  - Possibility of Inconsistencies

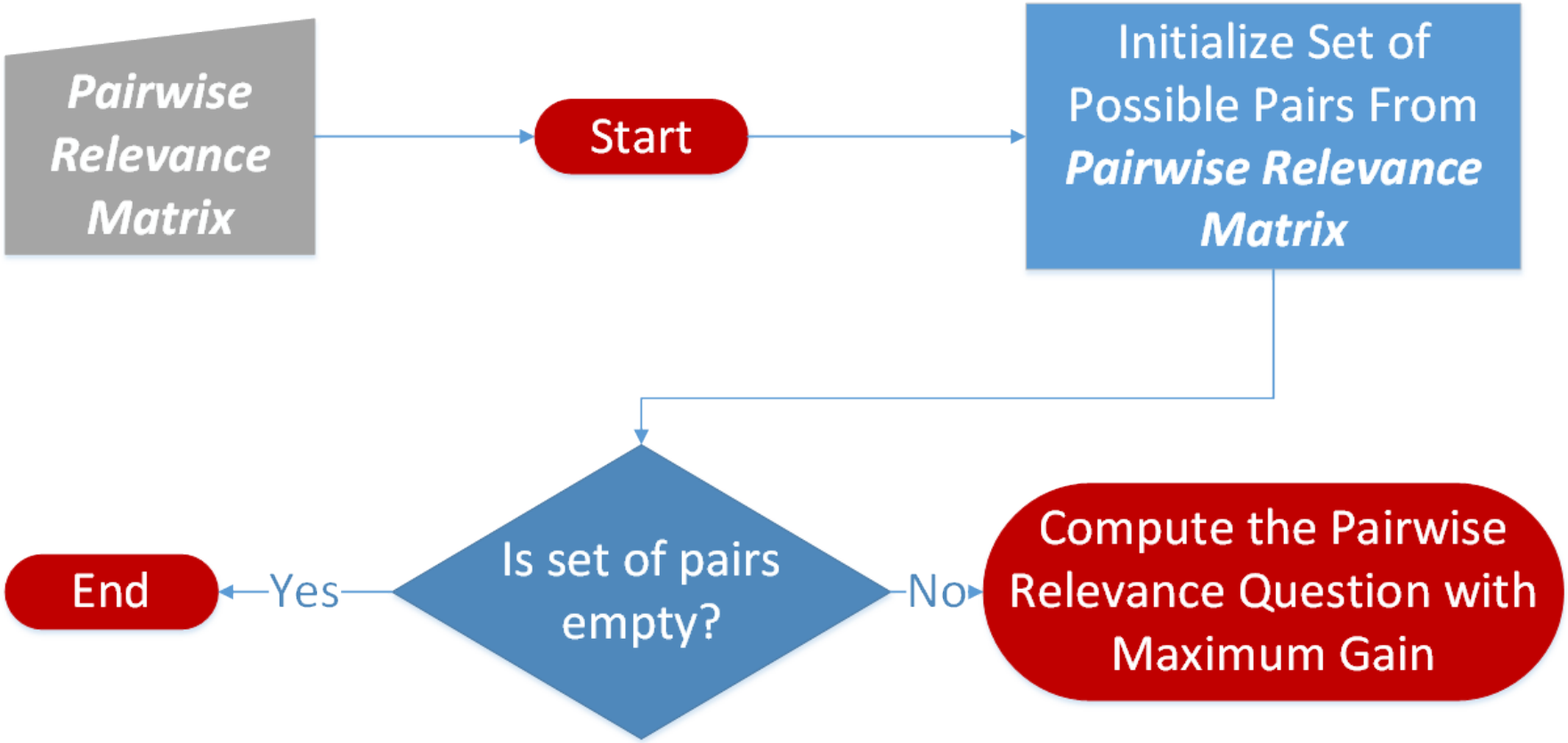
PointRank / Overview

- Parameters
  - *pois*: The list of PoIs to be ranked
  - *ina*: Initial number of assignments for each pairwise question
  - *minni*: Minimum number of iterations to check for consensus
  - *maxni*: Maximum number of iterations for each pairwise question
  - *pvalue*: Maximum p-value to consider the changes significant
  - *pt*: Probability threshold to decide about the answer of pairwise relevance questions
- Two phases:
  - Determining the Next Question
  - Processing the Question

PointRank / Overview

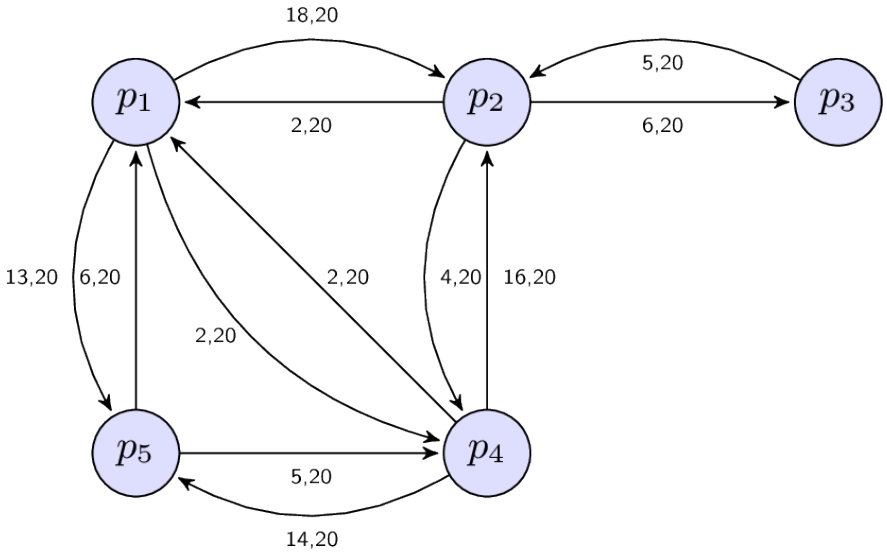


PointRank / Determine the Next Question



PointRank / Gain Definition

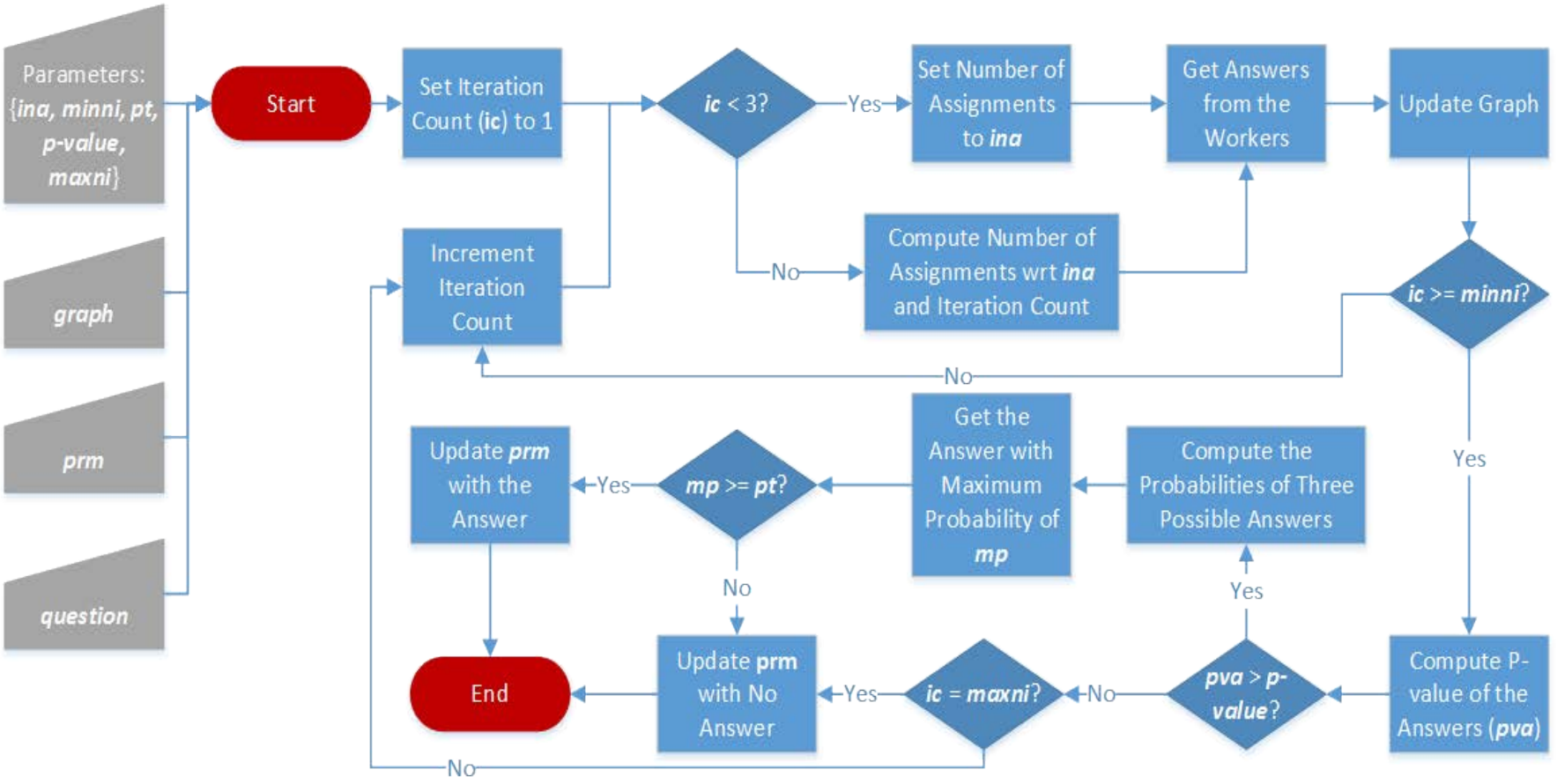
- Gain is defined as the number of pairwise relevance questions that may be eliminated by asking the question.
- $D = \{p_1, p_2, p_3, p_4, p_5\}$



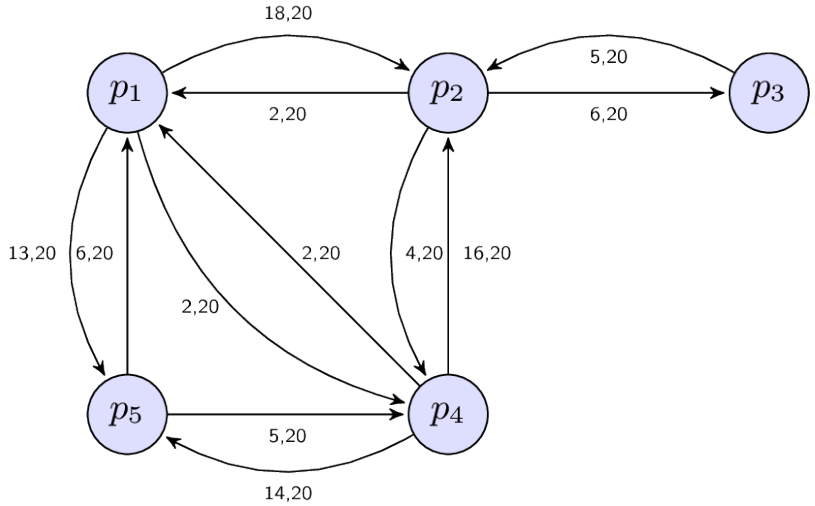
	p_1	p_2	p_3	p_4	p_5
p_1	0	1	2	0	1
p_2	-1	0	3	-1	2
p_3	2	3	0	2	2
p_4	0	1	2	0	1
p_5	-1	2	2	-1	0

- (p_3, p_4)
  - $p_3 \prec p_4$: From this answer, $p_3 \prec p_2$ and $p_3 \prec p_5$ can be inferred.
  - $p_4 \prec p_3$: No new pairwise relevance can be inferred.
  - Since the gain is defined as the average, the gain of this question is 1.

PointRank / Process the Question

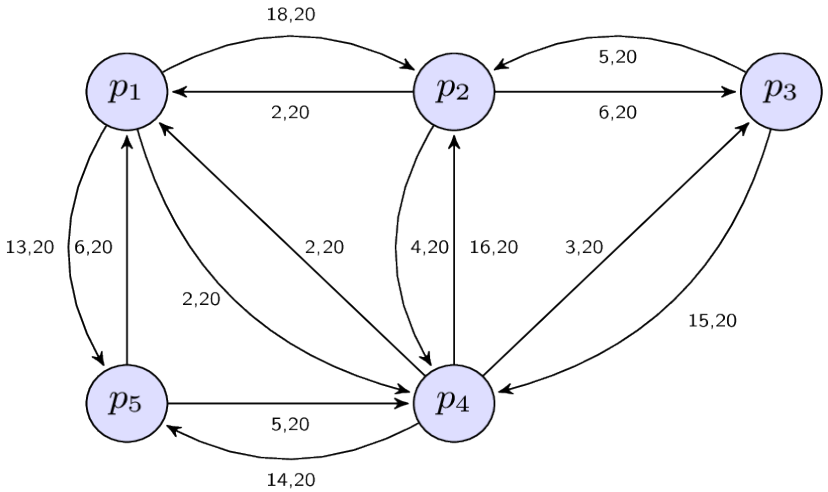


PointRank / Example (Cont'd)



	p_1	p_2	p_3	p_4	p_5
p_1	0	1	2	0	1
p_2	-1	0	3	-1	2
p_3	2	3	0	2	2
p_4	0	1	2	0	1
p_5	-1	2	2	-1	0

- Processing (p_3, p_4)
  - Parameters: *ina* = 5, *minni* = 3, *pt* = 0.6

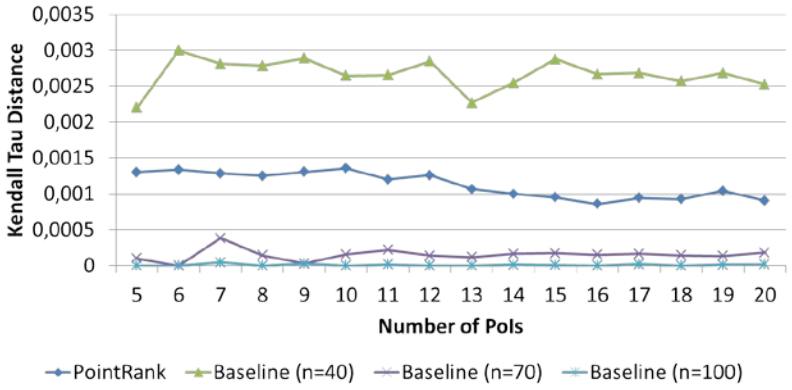


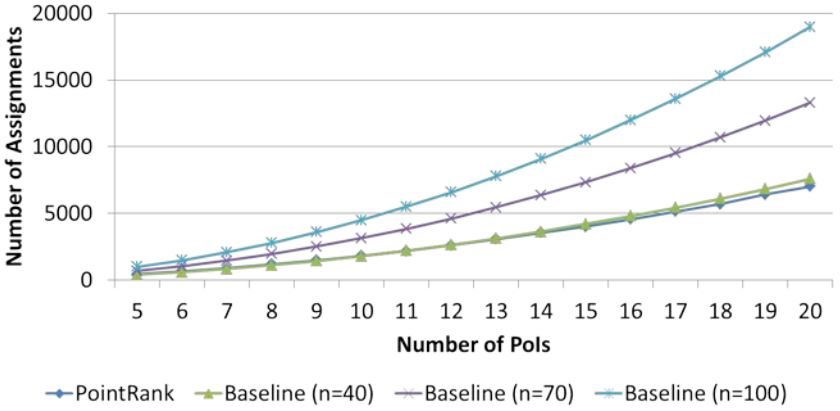
	p_1	p_2	p_3	p_4	p_5
p_1	0	1	2	0	1
p_2	-1	0	**-1**	-1	2
p_3	2	**1**	0	**1**	**1**
p_4	0	1	**-1**	0	1
p_5	-1	2	**-1**	-1	0

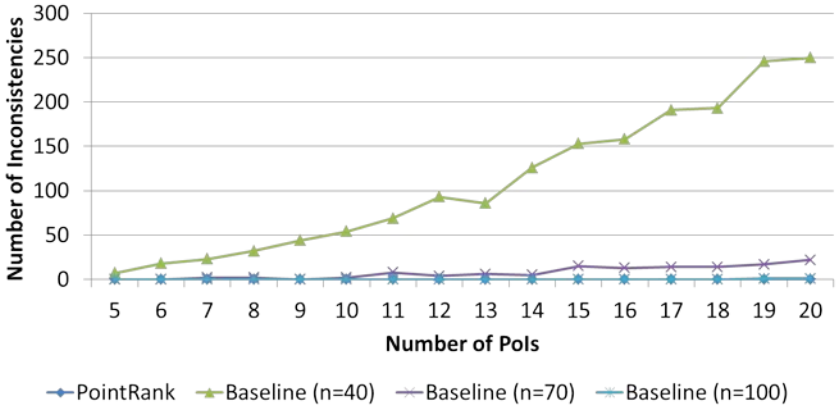
PointRank / Baseline Comparison

- Generated data
  - Total rankings
  - Partial rankings
- Baseline Algorithm
  - Majority Voting
    - Creates a fixed number of assignments about a question (*n*)
    - Determines the answer wrt majority
    - *n* = 40, 70 and 100
- Metrics
  - Kendall Tau Distance
  - Number of Assignments
  - Number of Inconsistencies
- Two main factors
  - Number of Places
  - Worker Reliability

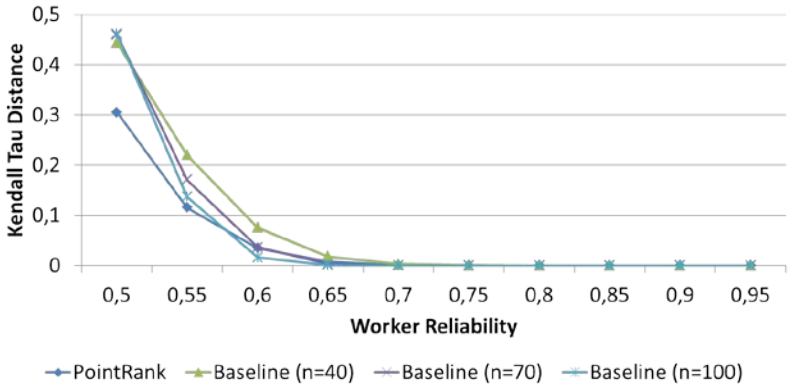
PointRank / Baseline Comparison

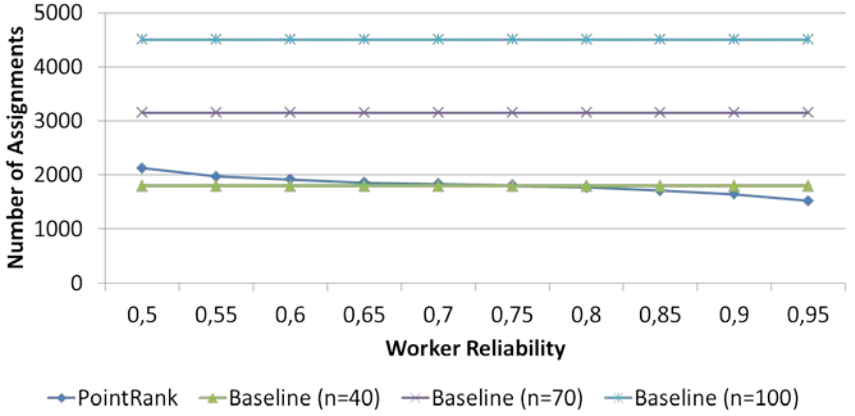


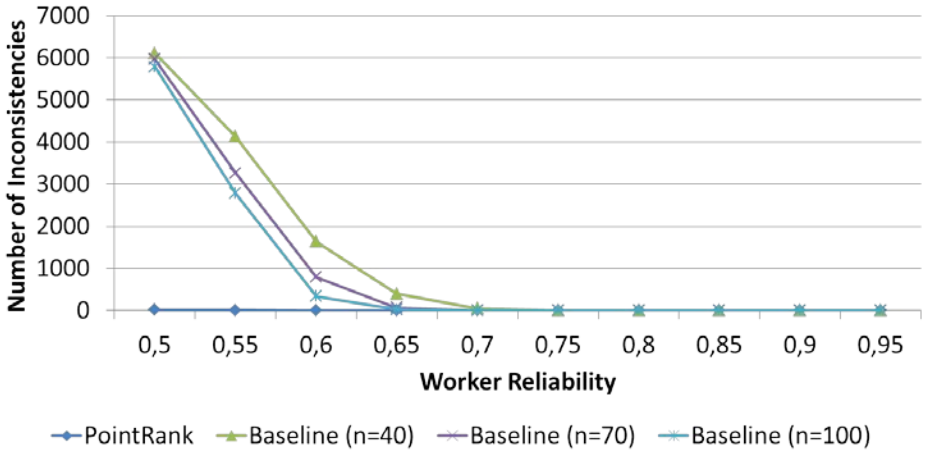




PointRank / Baseline Comparison







Conclusion

- We propose PointRank model
  - Synthesizes ranking of PoIs through crowdsourcing
  - Uses pairwise relevance questions
  - Is a step towards evaluation of ranking functions
- Evaluation methodology with synthetic data
- PointRank produces better results than an approach based on majority voting.