Technische Universität Darmstadt

Analyzing the Robustness of CertainTrust

Sebastian Ries, Andreas Heinemann

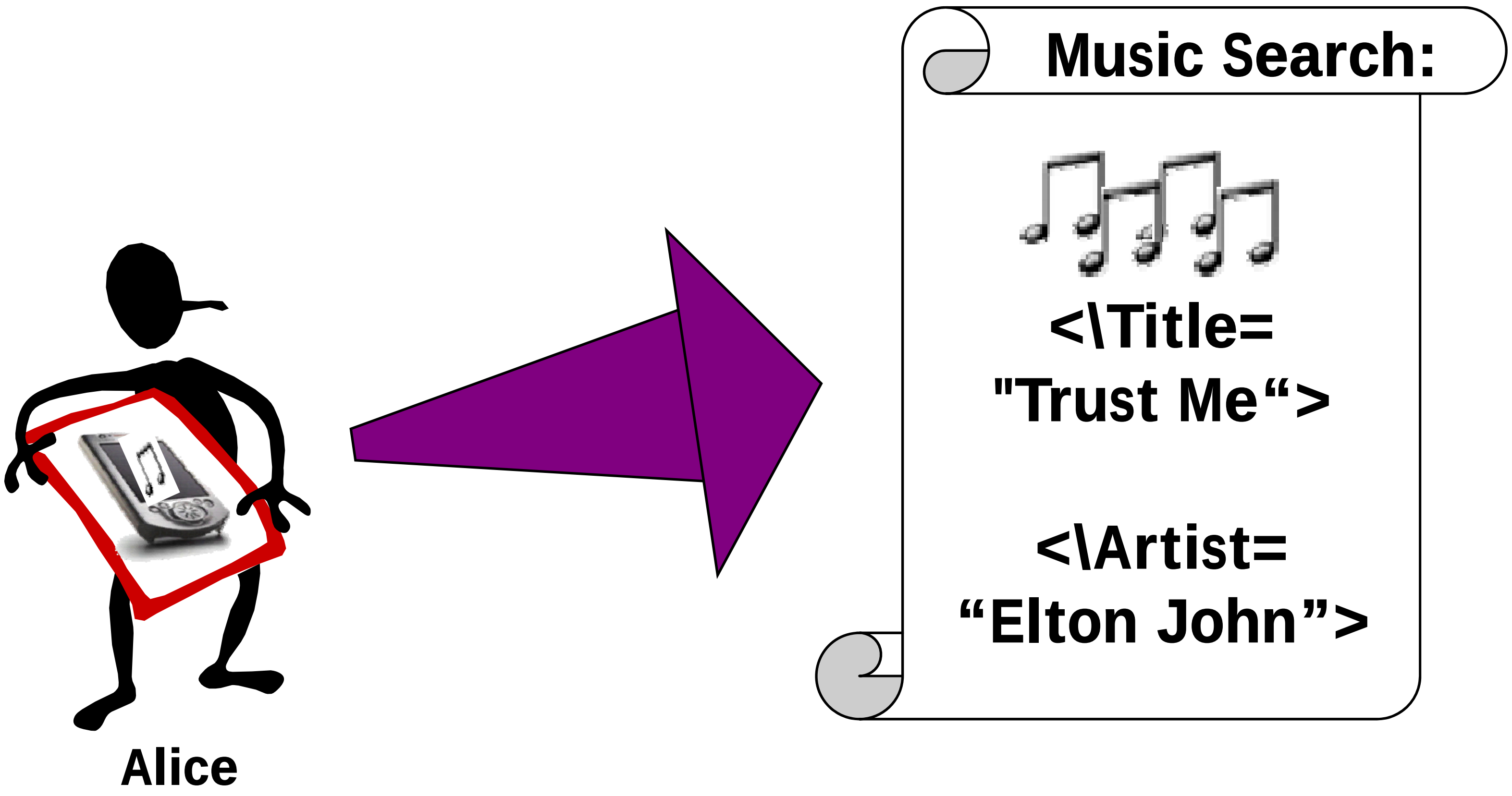
Deutsche Forschungsgemeinschaft DFG

Overview

- Motivation
  - What's the goal?
- Approach
  - CertainTrust: Deriving trustworthiness from evidence
- Robust integration of recommendations
  - Filtering, weighting & limiting
- Evaluation
  - What are the results?
- Conclusions
  - What we have achieved!

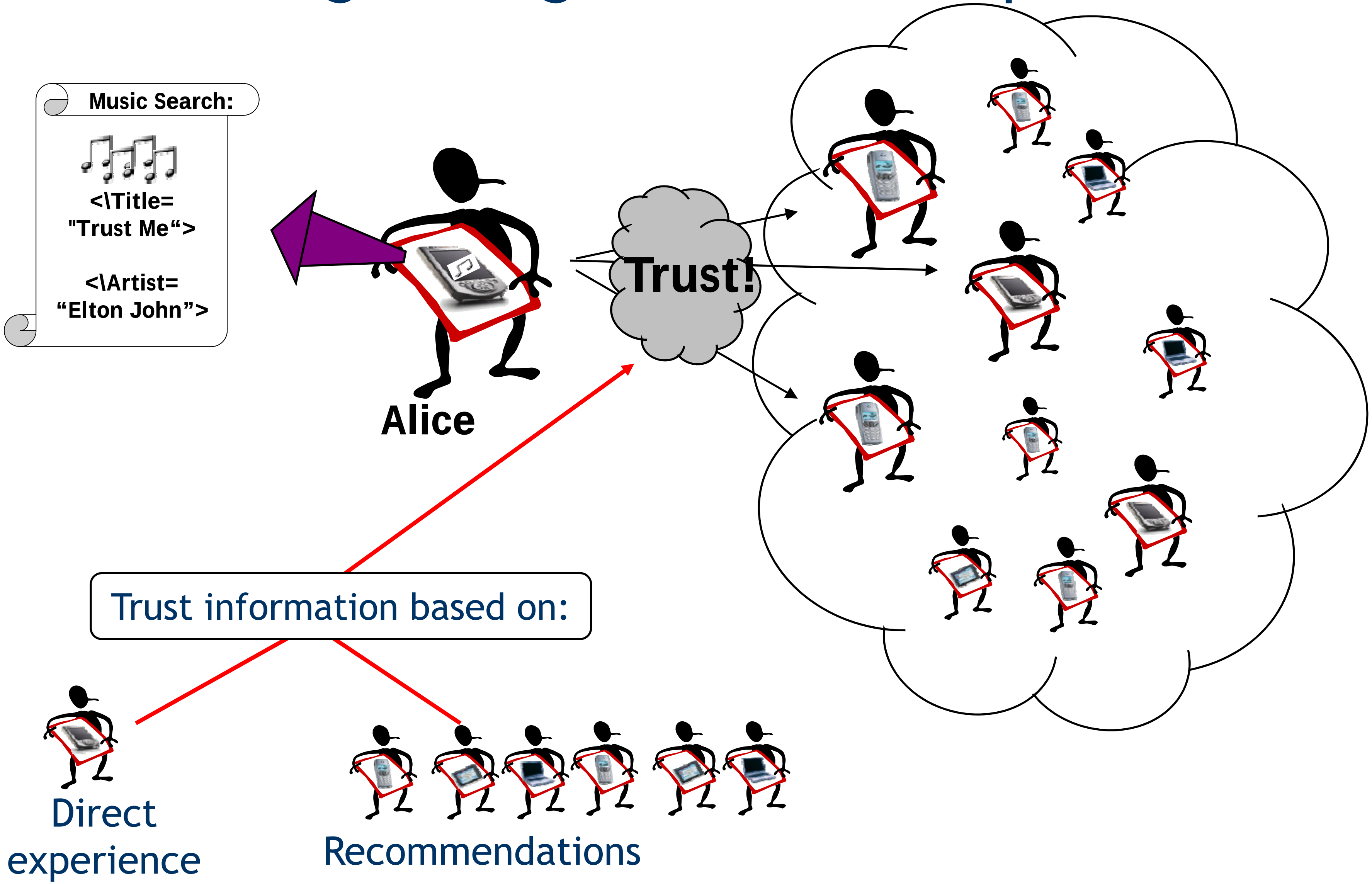
Motivation

Scenario: Collaborative Information Dissemination

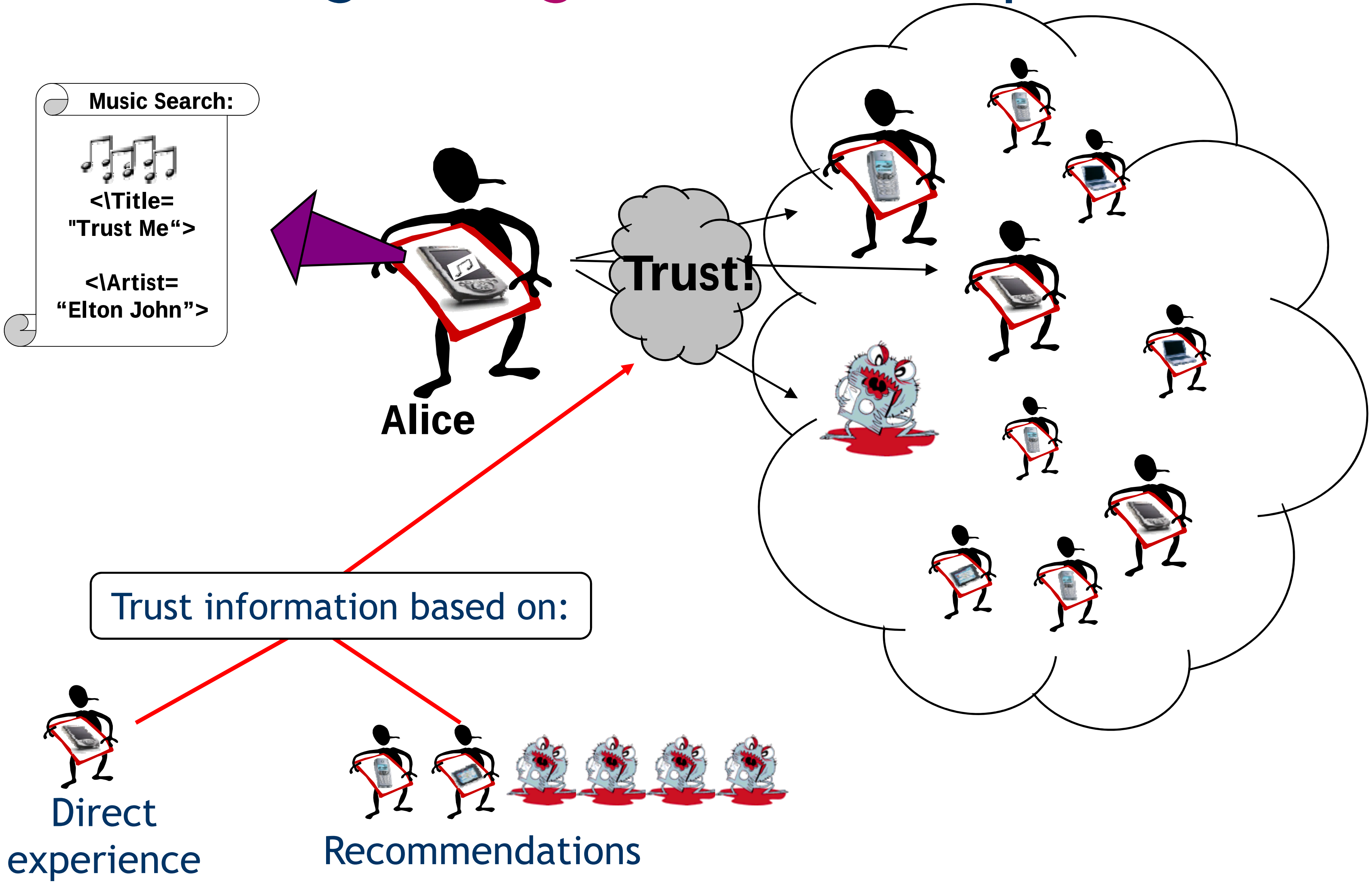


The crowd in front of the stadium

Finding the right interaction partner



Finding the right interaction partner



Beyond this simple scenario …

- Goal: Improving the quality of interactions using trust as a basis for decision making
  - Sub goal: Estimating the trustworthiness of an entity

- Approach: History based trust establishment using
  - Direct experience from past interactions
  - Indirect Experience: Recommendations
  - User knowledge

Approach

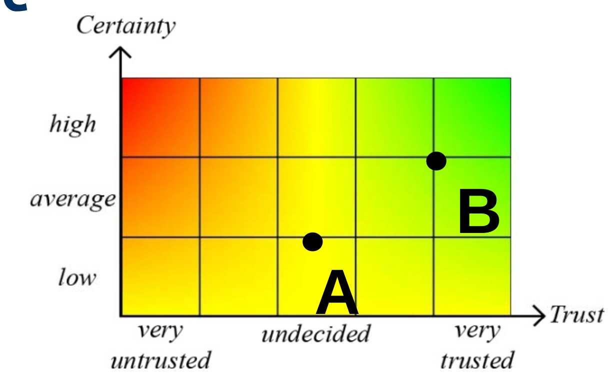
Challenge

- Definition: Trust (or, symmetrically, distrust) is a particular level of the subjective probability … [Gambetta]

- Estimating the trustworthiness of an entity
  - In the context of providing a service / interaction (e.g., file exchange)
    - Estimating the probability for providing a interaction with a positive outcome
  - In the context of recommendations
    - Estimating the probability for providing accurate recommendations

Both is done based on experience linked to past interactions

CertainTrust



- Experience (evidence):
  # positive / negative evidence

- Main parameters in the model:
  - Trust value (t):
    - Reflects the outcome of the past interactions
  - Certainty (c):
    - Increases with number of collected evidence
    - Limit for the collected evidence is maxExp
  - Initial Expectation (f):
    - The expectation about a positive outcome in an interaction with an unknown entity

$\Rightarrow$ Expectation Value (Estimated Trustworthiness) (E):
- E = t*c + (1-c) *f (alternatively bayesian mean)
- Subjective probability for positive outcome in the next interaction

How to choose the *initial expectation* f?

- Moderate approach: f = 0.5
  - Expectation value for an unknown entity is 0.5 (as assumed in most State-of-the-Art approaches)

- Selected alternatives: optimistic (f=1), pessimistic (f=0)
  - May also be appropriate! E.g., f=1 (or close to 1) in very friendly environments (or populations)

- Be aware this assumption may be wrong!
- Solution: Dynamically update f based on the experienced behavior over all encountered entities in the context!
  - Initial: value for f = 0.5
  - Learning based on encountered entities:
    - With positive experience => f shifts towards 1
    - With negative experience => f shifts towards 0

Robust integration of recommendations

Challenge

Overcome the problem
that direct experience may be rare …

… by aggregating …

- Direct experience
  - Incl. typical behavior of the community
- Recommendations

… in the face of lying recommenders

Approaches for robust integration of recommendations

- A recommendation is a tuple of pos./neg. exp:
  rec^A_B = (#pos., #neg.) – [A's direct experience with B]

- Filtering of recommendations
  - Consider only recommenders which provided mostly accurate recommendations

- Weighting of recommendations (Discounting)
  - Limit the number of evidence each entity may provide
  - Weight recommendations according to the trustworthiness of the recommender in the context providing accurate recommendations
    (using the right type of trust for recommenders!!!)

- Focus on direct experience and the best recommenders
  - Limit the number of evidence which is considered per interaction candidate

Achievements

- Gaining trust by recommendations is based on the most trusted recommenders:
  - Recommendations by unknown / little trusted recommenders have only small impact (if any)
  - Good resistance to attacks based on misleading recommendations

- Gaining direct trust requires …
  - for interactors: providing good interactions
  - for recommenders: providing accurate recommendations

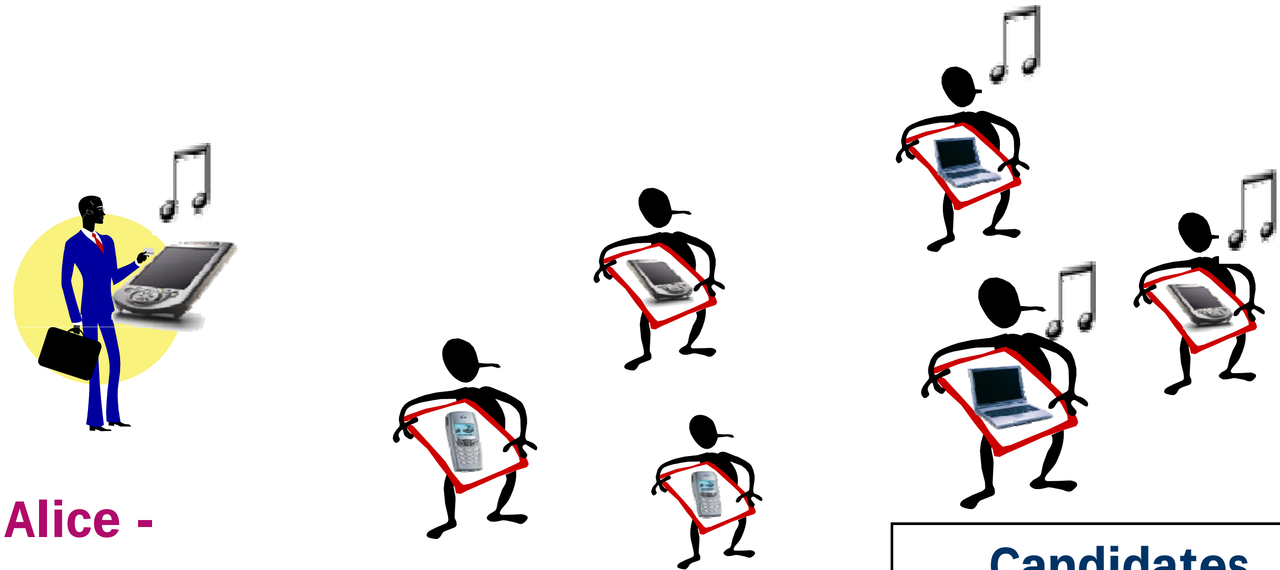
$\Rightarrow$ Attacks are connected to the costs of first providing good interactions or accurate recommendations

Evaluation

Evaluation

- Scenario: Collaboration in Opportunistic Networks
  - Users moving around with their personal devices sharing files with entities in proximity (distributed system)
  - Mobility model based on traces of the Reality Mining Project
    - About 100 participants, tracking based on mobile phones
    - Entities are assumed to be close to each other, if the are connected to the same mobile phone cell tower with a 15 min interval
  - Goal of an entity:
    Having as many interactions
    with positive outcome as possible!

Scenario – Select best interaction partner



Alice -

Initiator

Candidates

Recommenders

Selection based on direct experience & recommendations

Initiator MUST interact with best candidate

Behavior of entities

- Possible user behavior derived from system model

Basic entity behaviors		Recommendation behavior	
		+	-
Interaction behavior	+	honest (h)	selfish (s)
	-	malicious (m)	worst (w)

- Two settings modeling the stability of the interaction behavior
  - deterministic:

    P(“entity adheres to assigned interaction behavior”) = 1
  - probablistic:

    P(“entity adheres to assigned interaction behavior”) $\in_U$ [0.5;1]

Populations

- 15 canonically derived populations:
  - h, m, s, w, hm, hs, hw, …, hmsw

- Example:
  - Population h: all entities are honest
  - Population hm: 50% of entities are honest, 50% of entities are malicious
  - …
  - Population hmsw: 25% of entities are {honest, malicious, selfish, worst}

Baselines

- Random selection - (Const05)

- Distributed Variants of the BetaRepSys
  - Beta(_Simple)
    - No weighting of recommendations
  - Beta_D(iscounting)
    - Weighting of recommendations is based on the assumption that an entity's behavior as interactor is equal to its behavior as recommender

- Perfect Model
  - Doing the selection based on the knowledge of the true probabilities for positive outcomes

Evaluation Metrics

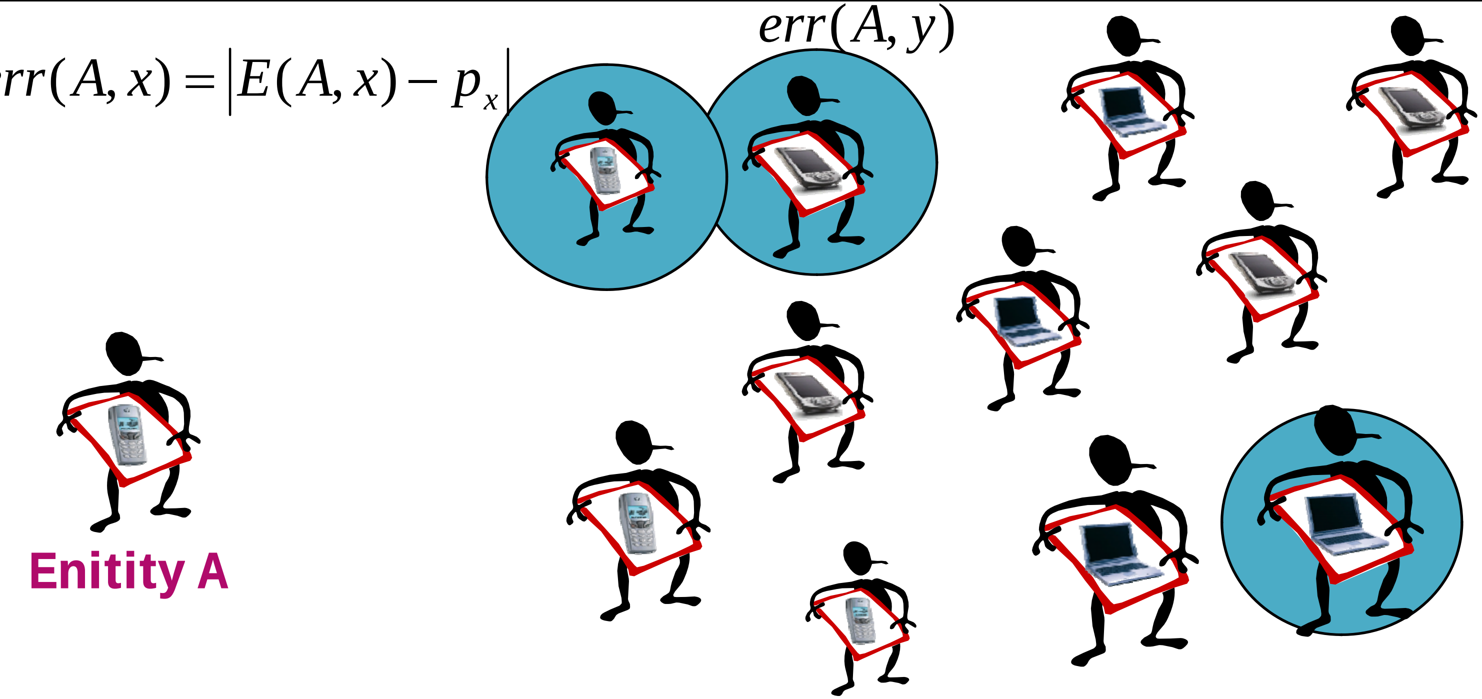
Error in Estimating the Trustworthiness

At the beginning of the simulation each entity is assigned a probability *p* for providing a "good interaction" (derived from the behavior).

The calculated trustworthiness is an estimate for this parameter.

$$err(A,x) = \left| E(A,x) - p_x \right|$$

$err(A,y)$



$err(A,z)$

Abs. Error Estimated Trustworthiness by Entity A:

$$err(A) = \sum_{x \in P} err(A,x) / |P|$$

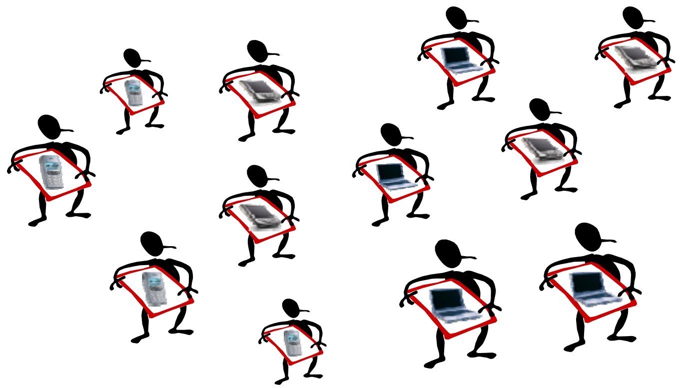
Avg. Abs. Error in Estimating the Trustworthiness for all entities =

(Abs. Error Estimated Trustworthiness by Entity 1

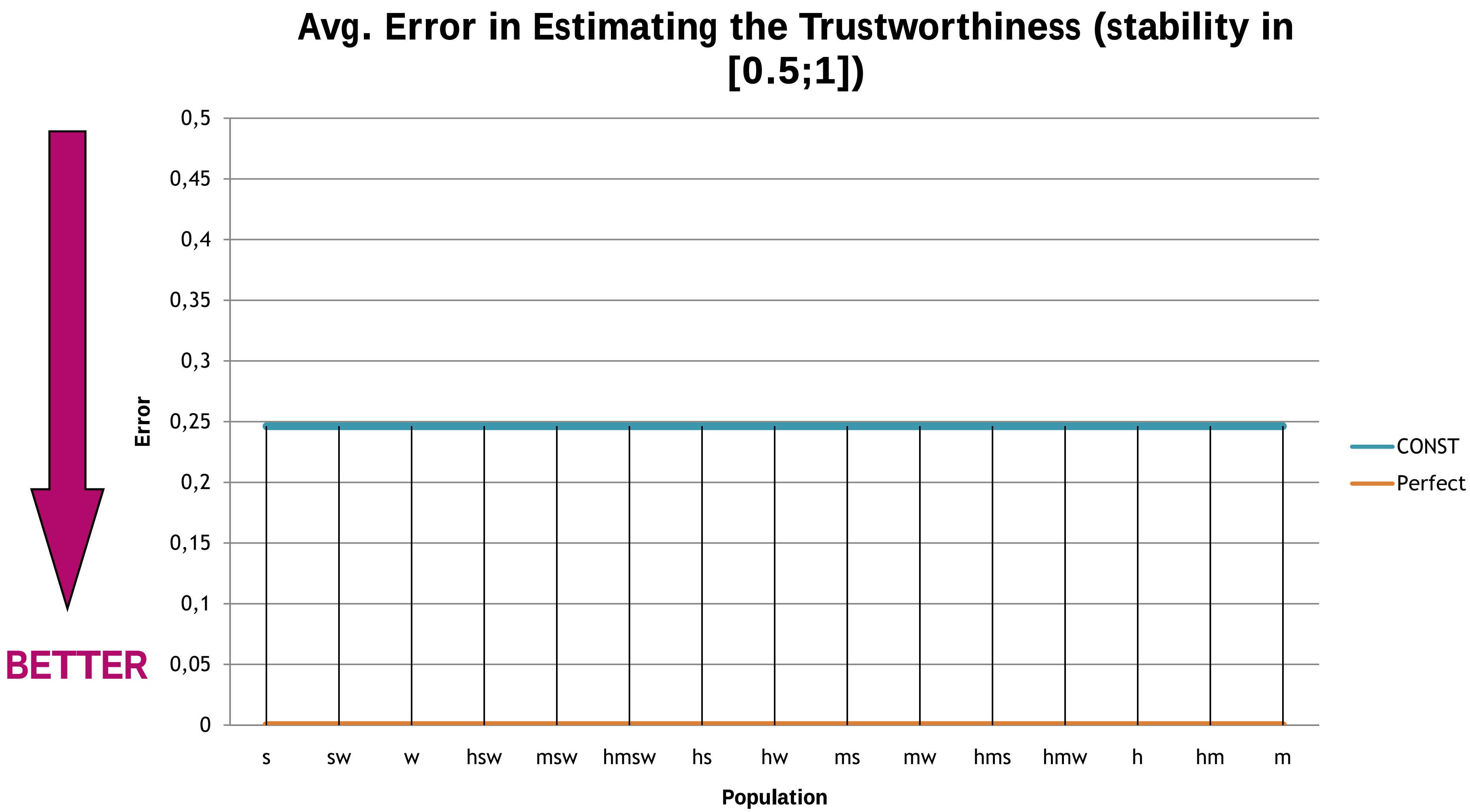
+ Abs. Error Estimated Trustworthiness by Entity 2

...

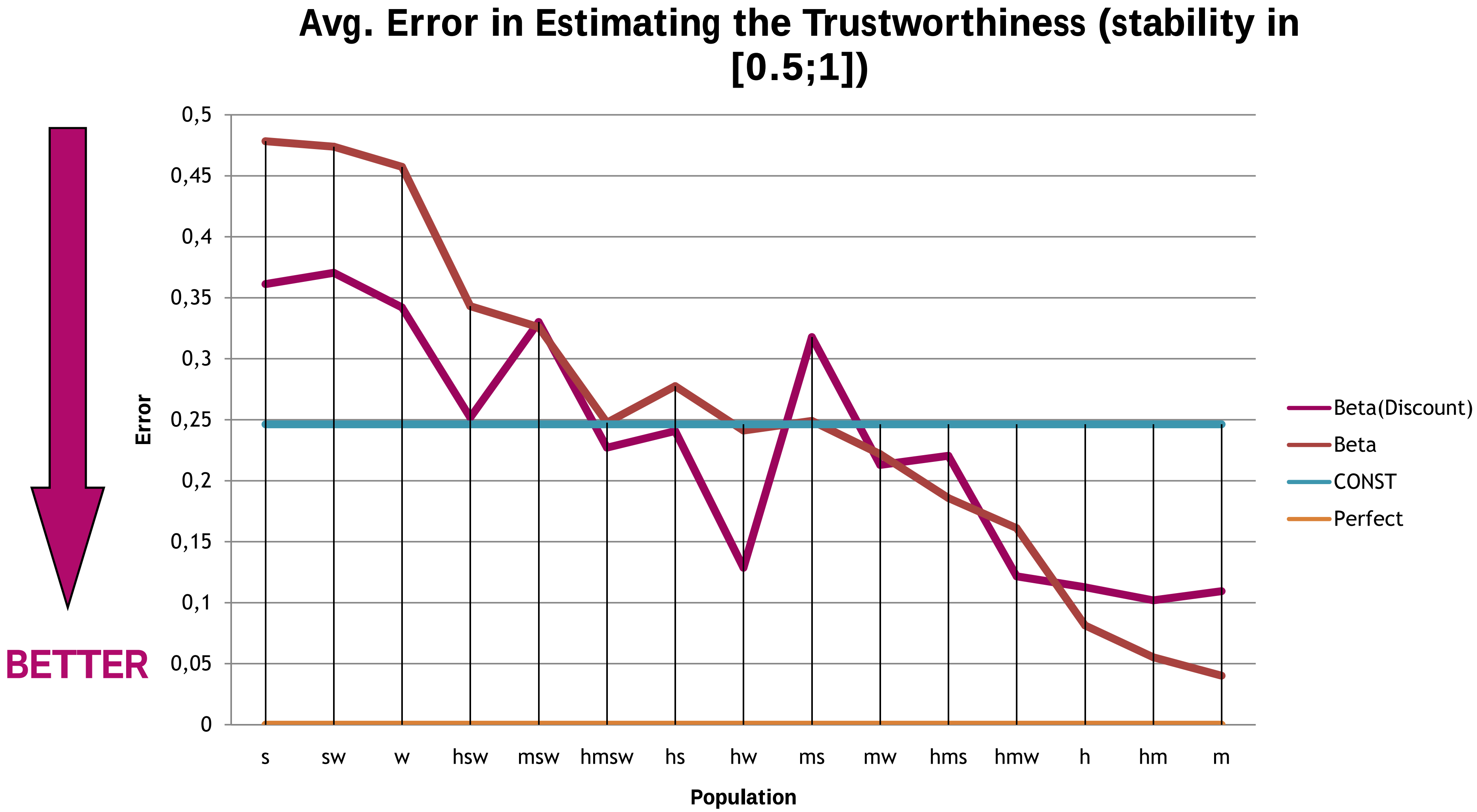
+ Abs. Error Estimated Trustworthiness by Entity n) / n



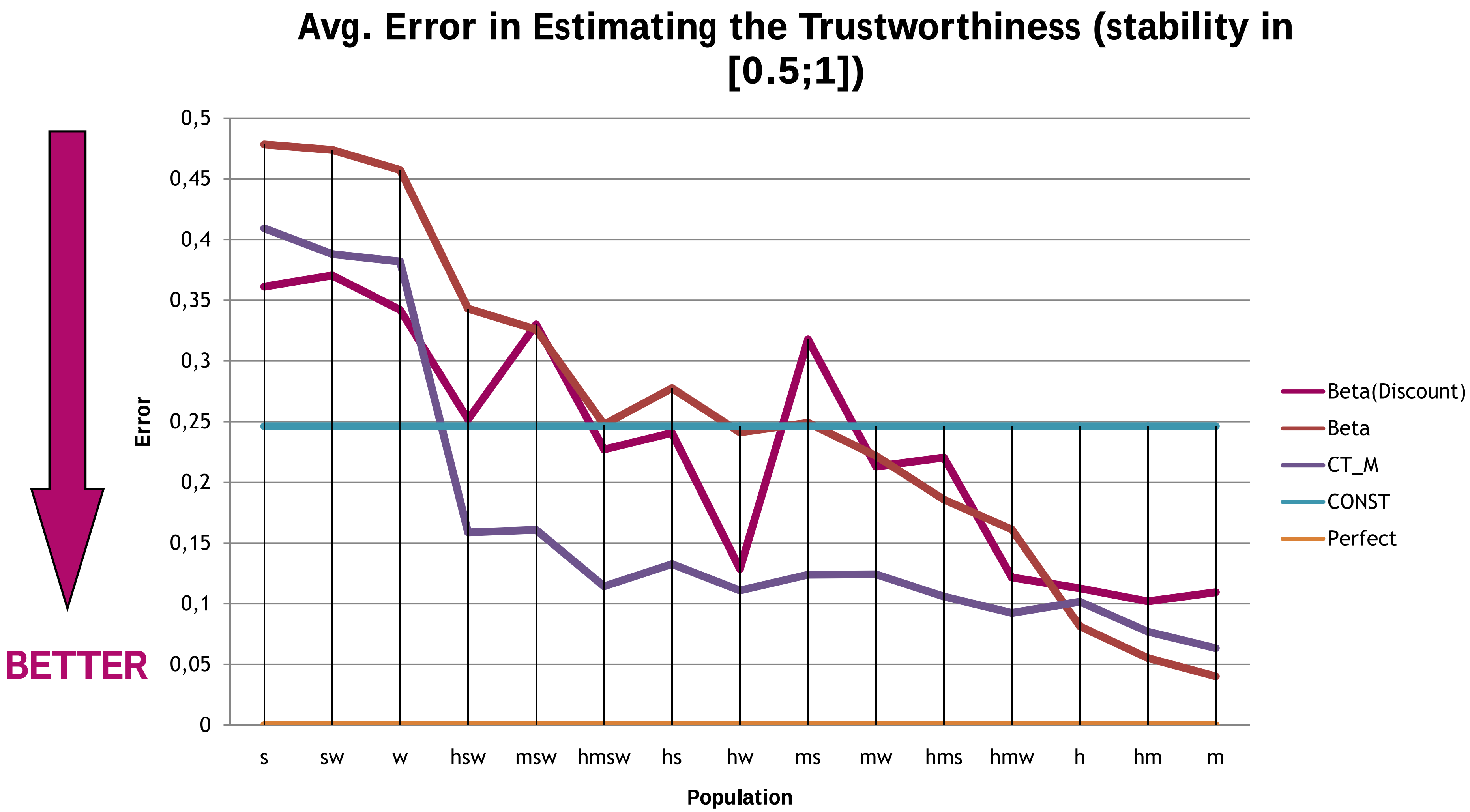
Results (trustworthiness)



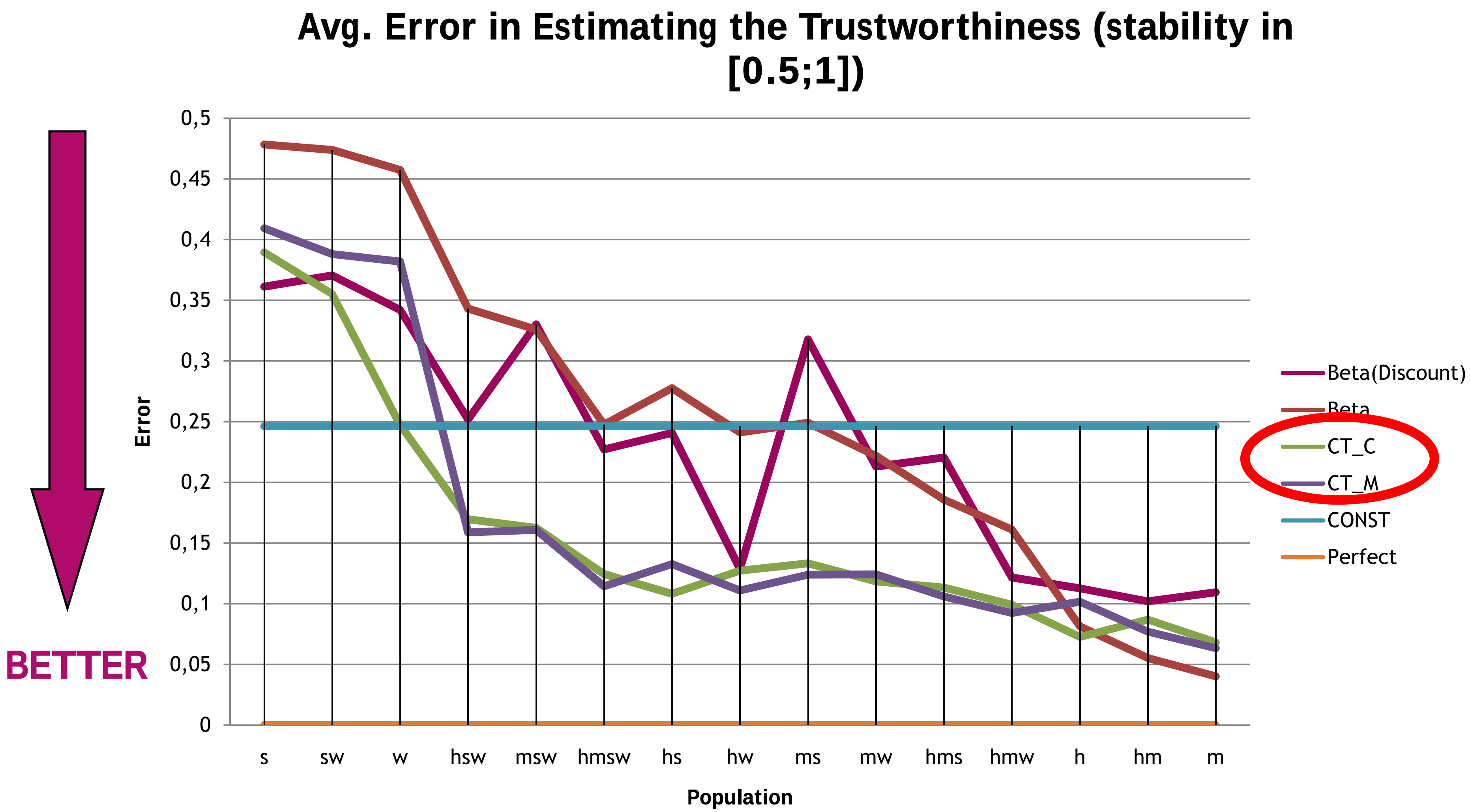
Results (trustworthiness)



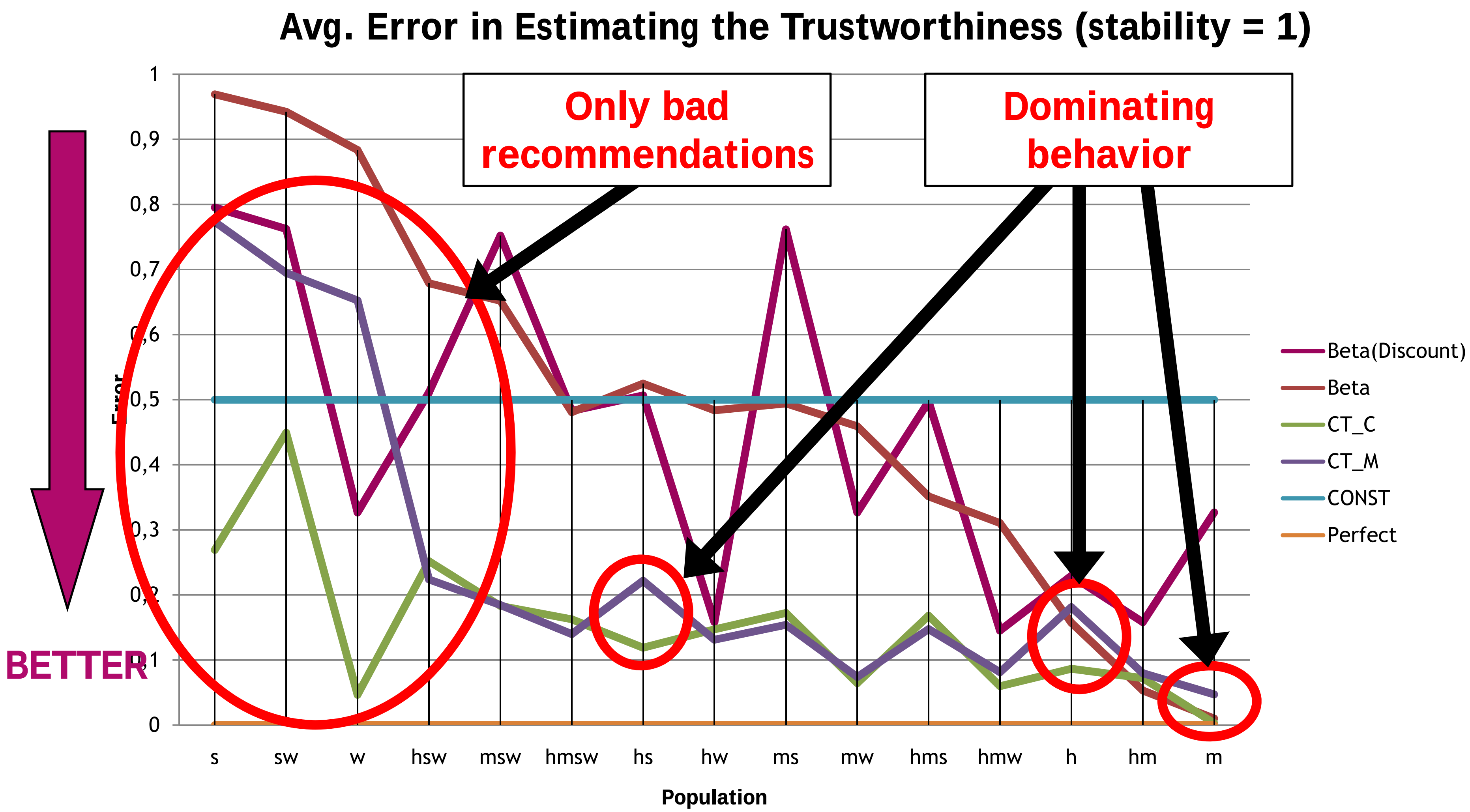
Results (trustworthiness)



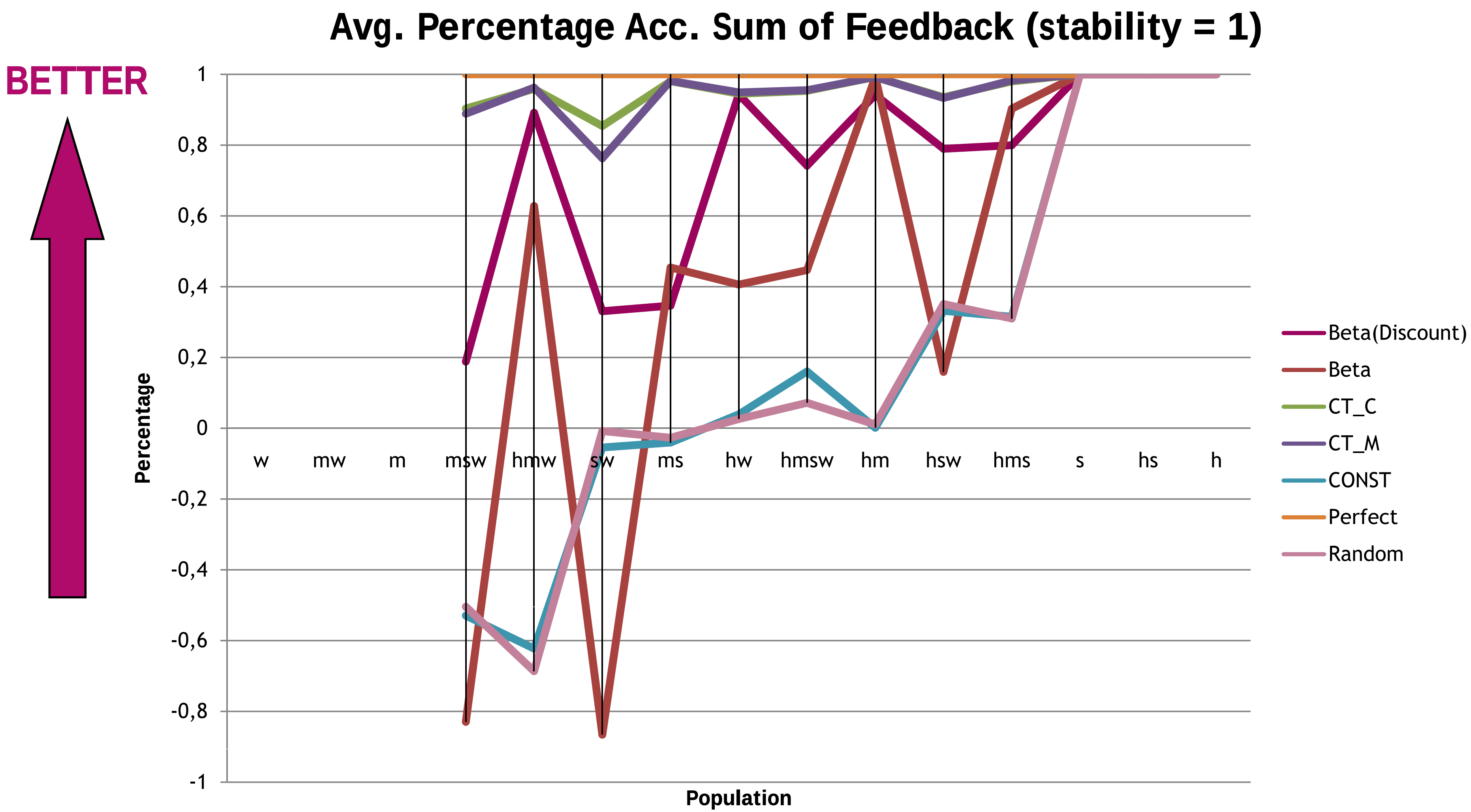
Results (trustworthiness)



Results (trustworthiness – stability = 1)



Results (acc. sum – stability = 1)



Interpretation of the Results

- Evaluated the model in a distributed scenario
  - Over a canonical set of populations
    - Showing the wide range of applicability of CertainTrust
  - With good results
    - Percentage acc. sum beyond 80% in 18 of 24 populations
    - Estimated trustworthiness allows to approximate probability of positive outcome

- The ideas should not be measured by the absolute numbers, but by the relative improvement!

Conclusions

- Provided a trust model
  - Allowing for dynamically updating the initial expectation about unknown entities
  - With robust filtering & trust update techniques
    - Limit influence of unknown/little trusted recommender
    - Using the right type of trust for weighting recommendations
    - Gaining direct trust is strictly linked to interactions
- Improved the overall quality of interactions
- Yet, enhancing robustness towards false recommendations and Sybil attacks beyond the simulated scenario

Thank you!