HOW TO

EVALUATE YOUR RESEARCH

Your Evaluation Section will consist of five parts:

GOAL	*”What’s the main goal of your proposal?"*
METRICS	*”What do you measure to ascertain whether your system meets the goal?"*
SETUP	*”What are the “ingredients” of your evaluation?"*
EXECUTION	*”Given the setup, how do you run your evaluation to measure the metrics?”*
RESULTS	*"So what did you get?"*

STRUCTURE

Every Evaluation Section has the same skeleton.

1	2	3	4	5
GOAL	**METRICS**	**SETUP**	**EXECUTION**	**RESULTS**
Goal of your proposal?	*What you measure?*	*How do you start?*	*How do you unfold?*	*What did you find?*

RUNNING EXAMPLE

Quercia, Hailes & Capra

"Lightweight Distributed Trust Propagation" — ICDM 2007

THE IDEA

A machine-learning algorithm that runs on a mobile phone and predicts how much user A should trust user B, given only a small subset of a network of trust ratings between users.

RUNNING EXAMPLE

(1) Look at Section 4 of this paper ...

Lightweight Distributed Trust Propagation

Daniele Quercia, Stephen Hailes, Licia Capra
Department of Computer Science, University College London, London, WC1E 6BT, UK
{D.Quercia, S.Hailes, L.Capra}@cs.ucl.ac.uk

Abstract

Using mobile devices, such as smart phones, people may create and distribute different types of digital content (e.g., photos, videos). One of the problems is that digital content, being easy to create and replicate, may likely swamp users rather than informing them. To avoid that, users may organize content producers that they know and trust in a web of trust. Users may then reason about this web of trust to form opinions about content producers with whom they have never interacted before. These opinions will then determine whether content is accepted. The process of forming opinions is called trust propagation*. We design a mechanism for mobile devices that effectively propagates trust and that is lightweight and distributed (as opposed to previous work that focuses on centralized propagation). This mechanism uses a graph-based learning technique. We evaluate the effectiveness (predictive accuracy) of this mechanism against a large real-world data set. We also evaluate the computational cost of a J2ME implementation on a mobile phone.*

RUNNING EXAMPLE

(2) Extract goal, metrics, setup, execution, results...

4 Evaluation

The goal of our algorithm is to predict trust ratings on portable devices. To ascertain the effectiveness of our algorithm at meeting this goal, our evaluation ought to answer three questions:
(1) *(Predictive Accuracy)* How accurate is our algorithm in predicting trust ratings?
(2) *(Prediction Robustness)* What is the impact of uncooperative users upon the algorithm's accuracy?
(3) *(Overheads)* What time, storage, and communication overheads does our algorithm impose on a mobile phone?

To see whether our algorithm effectively predicts trust and whether it is usable on portable devices, we need a *large-scale deployment*. Only so can we separate statistical significant answers from plausible insights gained by a small-scale deployment. Plus, a deployment needs to be evaluated in the *long-term* to see whether our algorithm is robust against, for example, uncooperative users.

Unfortunately, we do not have a long-term evaluation of a large-scale mobile computing deployment. We do, however, have a large rating data set from the Advogato community that has been around for more than a decade[5]. Using this data set (described next), we evaluate whether our algorithm is effective in predicting real trust ratings (Section 4.1). Then, to evaluate how robust our algorithm is, we emulate how users may rationally turn to be uncooperative (Section 4.2). Finally, we implement our algorithm to assess whether it is usable on a mobile phone (Section 4.3).

To begin with, let us describe the Advogato data set. Advogato is a community discussion board for free software developers. Using the Advogato's trust metric [15], each

<Examples of info extracted by students>

Goal: predict trust ratings on portable devices.

Metrics: Predictive Accuracy, Prediction Robustness, Overheads

Setup: Advogato dataset

Execution:

1. Get the dataset
2. Check whether the algorithm is effective in predictions for real trust settings
3. Emulate algorithm robustness
4. Asses feasibility of implementation on mobile

Results
(section 4.1, 4.2 and 4.3)

Goals

The goal of our algorithm is to predict trust ratings on portable devices.

Metrics

Accuracy, Robustness (as Fraction of unknown predictions)

Setup

Start with the Advogato data set, where each user has a single (global) trust value computed by composing other users ratings

Execution

we measure predictive accuracy by cross validation…

Results

shows that direct trust propagation performs better than naive prediction, ….

shows, if the number of uncooperative users reaches a critical point (if it is higher than 60%), …

METRICS:

(1) (Predictive Accuracy) How accurate is our algorithm in predicting trust ratings?
(2) (Prediction Robustness) What is the impact of uncooperative users upon the algorithm's accuracy?
(3) (Overheads) What time, storage, and communication overheads does our algorithm impose on a mobile phone?

SETUP:

1. Using this data set (described next), we evaluate whether our algorithm is effective in predicting real trust ratings.
2. Then, to evaluate how robust our algorithm is,
we emulate how users may rationally turn to be uncooperative.
3. Finally, we implement our algorithm to
assess whether it is usable on a mobile phone

EXECUTION:

- Cross Validation: Mask one trust relationship and then predict the relationship’s rating
- For each combination, we compute the predictive accuracy.
- Validation accuracy: computed the optimal parameters

RESULTS:

1)

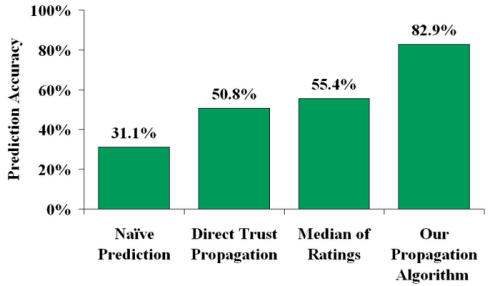


Figure 6. Predictive accuracy of four algorithms.

2)

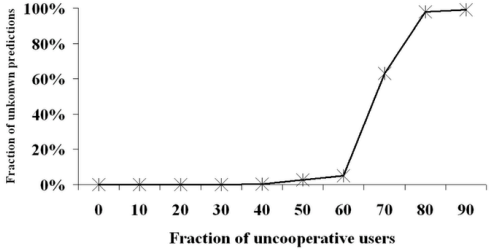


Figure 7. Fraction of unknown predictions as a function of uncooperative users (users who are not willing to make their ratings available).

3) Hence, say that the size of a tuple is roughly 10B. Even with 50 incoming and 50 outgoing edges (which is pessimistically high), the table size is 30KB. Also, for a single trust propagation, the data to be sent is less than 30KB.

The computation overhead, given as the mean of 10 runs, is as low as 2.8 milliseconds.

Evaluation:

1) The goal of our algorithm is to predict trust ratings on portable devices

Metrics:

(1) (Predictive Accuracy) How accurate is our algorithm in predicting trust ratings?

(2) (Prediction Robustness) What is the impact of uncooperative users upon the algorithm's accuracy?

(3) (Overheads) What time, storage, and communication overheads does our algorithm impose on a mobile phone?

<END of examples>

1 GOAL

WHAT'S THE GOAL OF YOUR PROPOSAL?

"The goal of our algorithm is to predict trust ratings on portable devices."

— Quercia et al., in one sentence.

One sentence, three sub-questions.

FROM A SINGLE GOAL,

three testable questions fall out:

IS IT ACCURATE?

Does it predict the right rating?

IS IT ROBUST?

What if some users won't cooperate?

IS IT CHEAP ENOUGH?

Can it actually run on a phone?

USE THIS TEMPLATE

"The goal of our [artifact] is to [do X] under [conditions]. To ascertain whether our [artifact] meets that goal, we ask:

(1) [Question 1]
(2) [Question 2]
(3) [Question 3]"

2 METRICS

WHAT ARE THE QUANTITIES YOU MEASURE?

A metric is a number that operationalizes an evaluation question

METRICS

Choose what you measure with care.

01

USE METRICS THE READER KNOWS.

"Predictive accuracy" means something to everyone. "Our novel trust-alignment score" means nothing to anyone.

02

GIVE A REFERENCE POINT TO EXPLAIN NUMBERS.

"82.9% accuracy" is meaningless until you add: random guessing scores 31%. That context turns a number into a claim.

DEFINE BASELINE

A baseline is a reference method you compare yours against. Without it, your number has no meaning. A baseline tells the reader "here is what we already knew how to do, and twe improved THAT much..."

THE FLOOR

Naive / random guess

Any non-stupid method must beat this.

THE SIMPLE ONE

Heuristic / median / mean

A cheap trick with no machinery.

THE STATE OF THE ART

Best published method

The real competitor you must surpass.

Quercia uses all three: random guess (31.1%), median-of-ratings (55.4%), direct propagation — the state of the art (50.8%).

3 SETUP

WITH THAT DO YOU START?

Typically, you should describe:

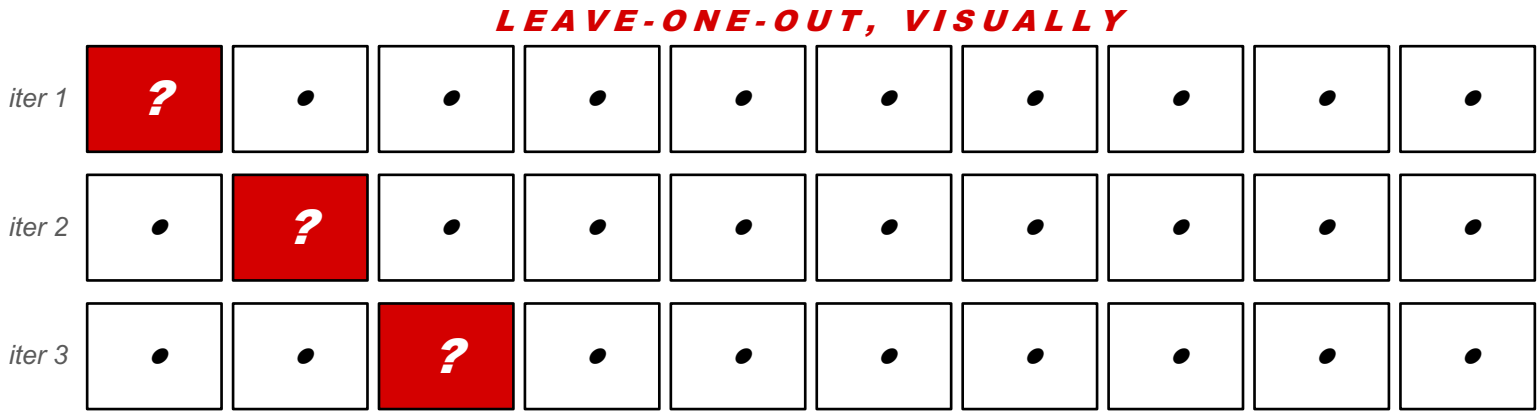
DATA	BASELINES	PARAMETERS	PLATFORM
Real or synthetic? How big? From where?	*What are you comparing against?*	*Every knob, every value — and why.*	*On what hardware? With what OS?*
EXAMPLE	EXAMPLE	EXAMPLE	EXAMPLE
Quercia: 55,455 trust ratings from Advogato.	Quercia: naive, median, prior state of the art.	Quercia: five parameters, 1,250 combinations.	Quercia: Nokia 3230, 123 MHz, Symbian 7.0.

4 EXECUTION

HOW DO YOU UNFOLD?

DEFINE CROSS-VALIDATION

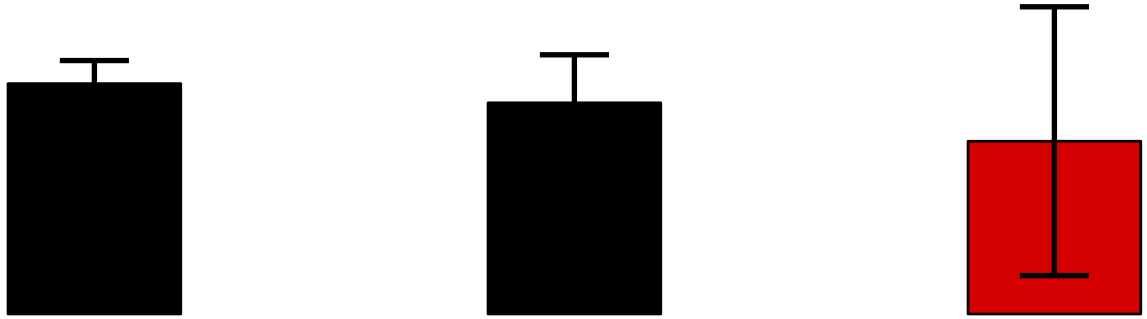
A trick to test a predictor when you have no "truth" other than your own data. Hide a bit of the data, train/predict on the rest, reveal the hidden bit, and check whether you got it right. Repeat.



Each row is one iteration. Hide one rating (red), predict it from the rest, score the prediction. Repeat for every rating. Average the score.

DEFINE CONFIDENCE INTERVAL

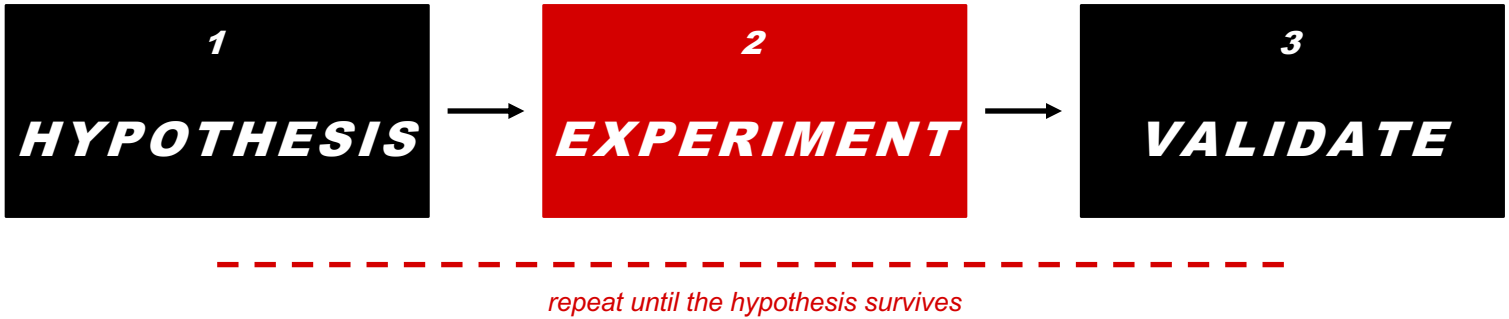
When you run a randomized experiment — anything involving shuffling, sampling, or a network — the single number you get is not "the truth." It's one draw. A confidence interval says: if we repeated the whole thing many times, the true value would fall in this range most of the time (usually 95%).



More runs ⟶ tighter interval ⟶ stronger claim. One run tells you almost nothing.

EXECUTION

Ideally, it takes the form of:



WRONG

"A has more overhead because it uses flooding." — You only showed overhead, not the cause.

RIGHT

"We hypothesize flooding causes the overhead. We lowered its frequency; overhead dropped to B's level. Confirmed."

5 RESULTS

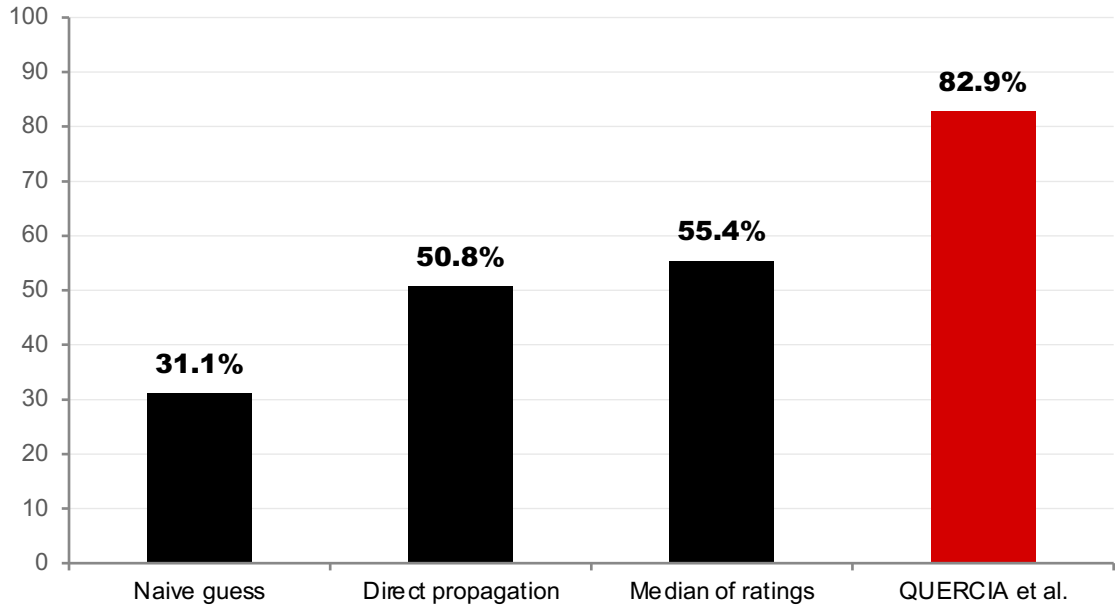
SHOW. **SAY.** **MEAN.**

Show the data. Say in words what it shows. Explain what it means.

Most student papers do step one and stop. That leaves the reader doing your job.

RESULTS · EXHIBIT A

One figure can carry a whole paper.



Quercia et al., Figure 6.

PREDICTIVE ACCURACY

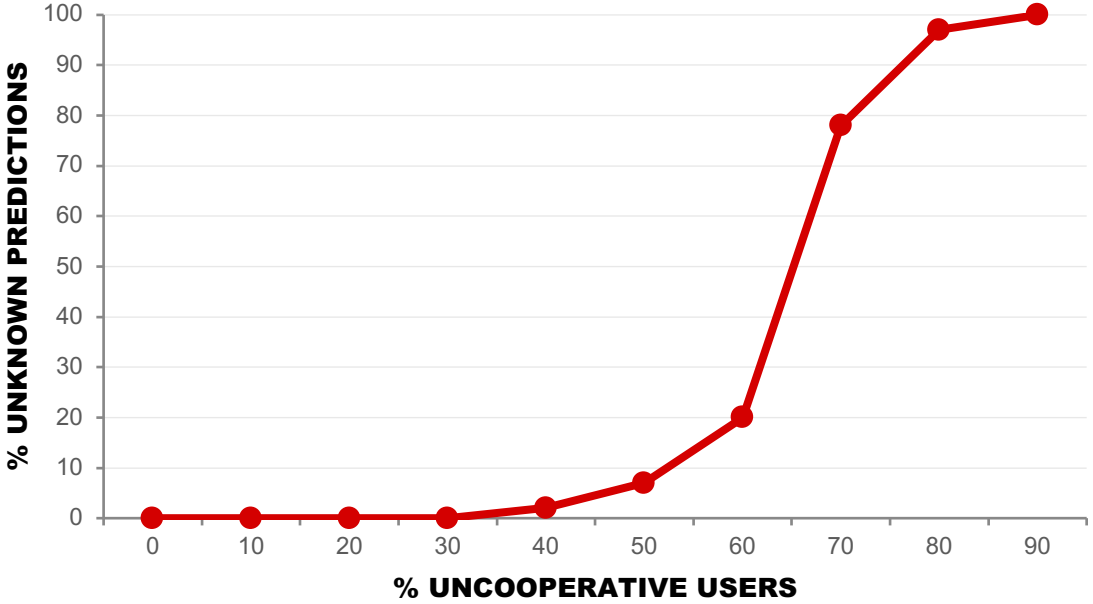
Four methods. One metric. The new method beats every baseline — AND we report the baselines, so you can judge for yourself.

82.9% vs 31.1% random

That gap is the contribution.

RESULTS · ROBUSTNESS

A phase transition — and why it happens.



Quercia et al., Figure 7.

THE PHASE TRANSITION

Up to ~60% uncooperative users, almost nothing breaks. Past that, the system collapses.

DON'T JUST DESCRIBE. EXPLAIN.

Quercia cites Albert et al. (Nature, 2000): scale-free social networks stay connected under random node removal — until they don't. Now the number means something.

Separate what you saw from what you think.

OBSERVATION

What the data shows.

"Accuracy drops sharply past 60% uncooperative users."

You can defend this from the graph alone.

INTERPRETATION

What you think it means.

"…because scale-free networks fragment beyond a certain removal threshold (Albert et al., 2000)."

This is a separate claim and must be supported separately.

Good writers signal the shift: "We observe… We hypothesize this is because…"

WAYS STUDENT PAPERS FALL APART

1

The "look how great" evaluation.

Cherry-picked benchmark, easiest baseline, flattering metric. Reviewers are trained to spot this.

2

No baseline, or a straw-man baseline.

If your only baseline is uniform random in a setting where nobody guesses uniformly, you've shown nothing.

3

A single data point.

One number on one workload on one machine is not evidence. Sweep, repeat, report variance.

4

Unexplained anomalies.

A weird-looking graph with no explanation makes readers suspect you don't understand your own system.

WAYS STUDENT PAPERS FALL APART

5

Claiming more than you measured.

You measured latency. You did not measure user happiness. Stick to what you actually measured.

6

Burying the headline.

The most important result goes in the first figure or the first sentence of §Results — not on page 9.

7

The "it just works" explanation.

When your system wins, explain why. When it loses, explain why. If you can't, you don't understand your system yet.

BEFORE YOU HAND IT IN

A ten-point checklist.

- [ ] 01 Does the Goal state specific, answerable questions?
- [ ] 02 Does every claim in your abstract map to a specific experiment?
- [ ] 03 Is each figure self-contained — caption, axes, units, legend?
- [ ] 04 Could someone reproduce your experiment from Section Setup alone?
- [ ] 05 Did you report baselines — one naive, one state-of-the-art?
- [ ] 06 Did you report variance — std dev, confidence intervals, or multiple runs?
- [ ] 07 Did you explain every anomaly in every graph?
- [ ] 08 Did you separate observation from interpretation?
- [ ] 09 Did you avoid claiming more than you actually measured?
- [ ] 10 Would a skeptical reader close the section convinced?

SOURCES

If you wish more:

A. Fox	*Hints for Technical Paper Writing (Stanford)*
M. Hanson & D. McNamee	*Efficient Reading of Papers in Science and Technology*
J.-Y. Le Boudec	*Writing a Paper — Please Check These Guidelines (EPFL)*
S. Peyton Jones	*How to Write a Great Research Paper (Microsoft Research)*
D. Quercia, S. Hailes, L. Capra	*Lightweight Distributed Trust Propagation (ICDM 2007)*
H. Schulzrinne	*Writing Technical Articles (Columbia)*
J. R. Shewchuk	*Three Sins of Authors in Computer Science and Math*
J. Widom	*Tips for Writing Technical Papers (Stanford InfoLab)*