TRULLO - local trust bootstrapping for ubiquitous devices

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—**Handheld devices have become sufficiently powerful that it is easy to create, disseminate, and access digital content (e.g., photos, videos) using them. The volume of such content is growing rapidly and, from the perspective of each user, selecting *relevant* content is key. To this end, each user may run a trust model - a software agent that keeps track of who disseminates content that its user finds relevant. This agent does so by assigning an initial trust value to each producer for a specific category (context); then, whenever it receives new content, the agent rates the content and accordingly updates its trust value for the producer in the content category. However, a problem with such an approach is that, as the number of content categories increases, so does the number of trust values to be initially set. This paper focuses on how to effectively set initial trust values. The most sophisticated of the current solutions employ predefined context ontologies, using which initial trust in a given context is set based on that already held in similar contexts. However, universally accepted (and time invariant) ontologies are rarely found in practice. For this reason, we propose a mechanism called TRULLO (TRUst bootstrapping by Latently Lifting cOntext) that assigns initial trust values based only on local information (on the ratings of its user's past experiences) and that, as such, does not rely on third-party recommendations. We evaluate the effectiveness of TRULLO by simulating its use in an informal antique market setting. We also evaluate the computational cost of a J2ME implementation of TRULLO on a mobile phone.**

I. INTRODUCTION

Using their handheld devices (e.g., PDAs, smart phones), users may produce, disseminate, and access digital content (e.g., news, photos, videos). In so doing, users may be engaged in everything from urban planning to creative expression [1] [2].

Digital content may span several categories (e.g., news may be about 'council politics' or about 'local events'), and only part of it is of high quality (defined subjectively - as, for example, interesting, relevant, accurate). To benefit from a wide variety of content, users should be able preferentially to select that fraction of content that is in their categories of interest and that is of high quality. Since the calculation of quality is subjective, each user should select content producers independently. To this end, a user may run a distributed trust model - a software agent that keeps track of who has (and who has not) disseminated content that its user has found relevant [3] [4] [5] [6]. Given that there are a myriad of (mobile) producers in a broad variety of categories, the following problem arises: before trust models can be used effectively, a critical mass of initial trust values need to be *effectively* set;

the exact amount of such values dramatically increases with the number of producers and of content categories (contexts[1]). This problem has been extensively discussed by Toivonen *et al.* [9].

Current research literature suggests that user A may assign its initial trust in user B:

- as a constant representing A's initial disposition to trust [10];
- based on other users' recommendations about B [11];
- close to A's trust in B in a similar and known context [12].

We propose a solution complementary to those existing approaches and discuss which of these approaches to use in what case (Section VIII). In so doing, we make the following contributions:

- Designing a novel mechanism, named TRULLO (TRUst bootstrapping by Latently Lifting cOntext), that statistically analyzes the ratings of its user's past experiences to bootstrap unknown trust values. It does so through a technique called Singular Value Decomposition [13]. In contrast to existing approaches (discussed in Section IX), TRULLO relies only on *local* information (the ratings of its user's past experiences) and, as such, does not need to collect recommendations, thus avoiding the need for a common ontology shared by all (recommending) users. Section IV-A introduces the key ideas on how TRULLO bootstraps trust and Section IV-B then proposes an algorithm implementing those ideas.
- Evaluating the effectiveness of TRULLO at predicting trust (Section V) by simulating its use in informal antique markets (described in the next section).
- Evaluating the computational cost of a J2ME implementation of TRULLO on a Nokia 3230 mobile (Section VII).

II. RUNNING EXAMPLE

Throughout the paper, we consider an application of handheld devices in informal antique markets. Given that those markets are huge, one major problem is that visitors cannot see prices of everything in which they are interested. To solve that problem, stall holders and visitors could use their handheld devices, such as their mobile phones and PDAs. Stall holders may, for example, disseminate ads of the items they are selling

---

[1]In the literature of trust management, content categories are called contexts (see, for example, [7] and [8]).

through their handheld devices, and visitors may collect those ads upon which they may compare prices and consequently decide which stalls to visit. Of course, visitors would have time to read only a subset of the ads. As such, they would wish to receive ads that they personally find *relevant*.

More specifically, consider visitor A.

- A receives ads from sellers in various contexts[2].
- The *relevance* of each ad is rated. By relevance we mean a (subjective) measure of satisfaction that a user gets from an ad. User A does not have explicitly to rate an ad's relevance: monitoring the time A spent reading the ad, or being physically located in front of the stall that is the subject of the ad, might serve as surrogates for rating.
- On input of those ratings, A's trust model updates its trust values for sellers and arranges those values in a seller-by-context matrix.

Now, consider that A is interested in ads in context c_x, and consider that a collection S of sellers sends ads in that context. Based on the values of A's seller-by-context matrix, TRULLO extracts statistically relevant information that we call "features"; by combing those features, TRULLO then assigns initial trust values to the sellers in S, selects the most trustworthy and pass only their ads to A.

III. ASSUMPTIONS

Before describing TRULLO, we spell out our assumptions:

- A mechanism to distinguish one context from another is given. This mechanism does not need to describe the relationships among all possible contexts (as does an ontology tree); rather, it simply needs to distinguish contexts. An example of such a mechanism is a directed graph, whose nodes are contexts and whose edges are change conditions [14]. A qualitative context description (e.g., 'Roman coins') may be then mapped into a node of the graph by considering that change conditions specify whether certain keywords are present in the description. This is a particular example of how to distinguish contexts, but each device that runs TRULLO is free to choose any other. This is an advantage of TRULLO's bootstrapping compared to bootstrapping from recommendations: while the latter requires that all recommenders and recipients of recommendations differentiate contexts in the same way, TRULLO simply expects each device to have *its own way* of doing so.
- TRULLO is given a way to distinguish one seller from another. That is, it must be able to *uniquely* identify sellers (e.g., by binding together a unique public key with each seller). If identities are not unique, TRULLO may suffer from sybil attacks [15], in which a malicious seller takes on multiple identities and pretends to be multiple, distinct sellers. In the absence of a central authority, TRULLO may statistically guarantee seller-identity bindings by using a mechanism similar to SybilGuard [16] or by using threshold cryptography [17].

[2]By context, we simply mean a textual description of an ad's category (e.g., 'Roman coins').

Fig. 1. Singular value decomposition (SVD). On a visitor's device, TRULLO gathers known trust values in a seller-by-context matrix D, upon which it applies the decomposition. Based on the three resulting matrices, it then reconstructs the trust value to be bootstrapped. The resulting matrices are: seller-by-feature matrix E (trust in sellers across features), diagonal feature matrix F (feature contribution to trust assessment), and transposed context-by-feature matrix G^T (feature relevance across contexts). To ease illustration in Section IV-B, some elements are in bold or underlined.

IV. TRULLO BOOTSTRAPPING MODEL

If those assumptions hold, TRULLO is able to bootstrap unknown trust values by applying the Singular Value Decomposition (to which we will refer as SVD and 'the decomposition' interchangeably) [13]. We introduce the decomposition first (Section IV-A), and then look at how we will apply it to the problem at hand (Section IV-B).

A. Singular Value Decomposition

To informally understand the main idea behind the decomposition, consider that we have a set of *known* trust values (resulting from past experiences) and have one *unknown* value to be initialized. The decomposition expresses the known values as linear combination of numeric "features". Then, those features are combined to set the unknown value. The nice property is that numeric features are statistical properties that are not specified by users but are automatically "intuit" by the decomposition. That is why TRULLO bootstraps trust *latently*.

More formally, consider a visitor (Alice) and its (n_s x n_c) trust value matrix D in which n_s rows represent the sellers from whom Alice has previously received ads in n_c contexts. For example, each element D_{iy} represents Alice's trust value in Bob (the ith seller) for sending ads about 'Roman coins' (the yth context). Applying SVD on the matrix D means decomposing it into three other matrices (see Figure 1):

$$D_{n_s \times n_c} = E_{n_s \times n_f} \cdot F_{n_f \times n_f} \cdot G^T_{n_f \times n_c},$$

where:

- Each E's i^{th} row represents a set of Alice's trust values in the ith seller across *features* - for example, E_{ik} represents Alice's trust value in Bob (the ith seller) in the kth feature.
- F's diagonal contains all features' contributions to trust assessment arranged in *descending* order[3] - for example, F_{kk} expresses the extent to which the kth feature impacts on Alice's assessment of trust.
- Each G^T's k^{th} row reflects the relevances of the kth feature across contexts - for example, G^T_{ky} represents the relevance of the kth feature in the yth context 'Roman coins'.

Assume, for the sake of argument, that we do not know the trust value for the ith seller in the yth context. From the three decomposed matrices, we may initially estimate that value as the *weighted* combination of trust values in the ith seller across *all* features. For each kth feature, the weighting factors are two: the feature's relevance in the yth context (G^T_{ky}), and the feature's influence on trust assessments (F_{kk}). For example, Alice's initial trust value in Bob for sending ads of 'Roman coins' equals the *weighted* sum of Alice's trust values in Bob in all features. Each feature is weighted according to its relevance in the context 'Roman coins' and to its general influence when assessing trust. Importantly, we do not need to explicitly define features: from trust values, the decomposition extracts statistically relevant information (features) and assigns, to each of the features, a number corresponding to its statistical relevance. TRULLO then latently (no context ontology required) bootstraps trust.

B. Bootstrapping Steps of TRULLO

Knowing how the decomposition can be applied to reconstruct unknown trust values, we are now ready to devise the bootstrapping steps of TRULLO. To do so, we consider the case in which TRULLO running on A's device has to bootstrap a trust value t for B in context c_x ('Roman coins'). To bootstrap t, TRULLO carries out the following steps:

Step 1. Determine the contexts in which A has previously received ads from B. For example, consider a situation in which those contexts are c_w, c_y, and c_z.

Step 2. Determine the sellers from whom A has received ads in the previously identified contexts plus that to be bootstrapped (i.e., in c_w, c_x, c_y, and c_z). For example, those sellers may be C, D, and E. Overall, TRULLO considers n_s sellers in n_c contexts. In this example, $n_s = n_c = 4$; in general, $n_s \neq n_c$. In the extreme case of TRULLO having limited information (i.e., either n_s or n_c equals 1), TRULLO bootstraps t depending on the following cases:

  1) $n_s = 1$ and $n_c = 1$ (A has received no ad - TRULLO has just been installed on A). TRULLO has no information, but has at least one default recommender: its user. So TRULLO

bootstraps t according to its user's risk attitude, which existing mechanisms (e.g., [19]) may elicit.

  2) $n_s = 1$ and $n_c > 1$ (A has received ads only from B and has done so in n_c contexts). TRULLO has to formulate a hypothesis of "how trustworthy B is" in a new context. It does so by setting t as the median of its known trust values in B. We choose the median instead of, for example, the mean because the median gives less weight to outliers [20].

  3) $n_s > 1$ and $n_c = 1$ (A has received ads from n_s sellers in the bootstrapping context, but not from B). TRULLO bootstraps t based on social investigations into how humans set initial trust in the real world. In the most-cited model of generalized trust by Hardin [21], initial trust is based on trusting disposition, which, in turn, is based on accumulated experiences *within* a particular context. In this vein, TRULLO bootstraps t as the median of the trust values for the sellers known in the bootstrapping context, which may be interpreted as the 'typical behavior' for sellers in that context.

In the above cases (very limited information), a simple median may appear to be a reasonable choice. However, in the presence of enough information ($n_s > 1$ and $n_c > 1$), Section VI on "Simulation Results" demonstrates that the results obtained by extracting features are by far more accurate than simply using the median (even for small n_s and n_c). The next steps detail how TRULLO extract those features.

Step 3. Populate a matrix $D_{n_s \times n_c}$. In our example, this matrix contains Alice's trust values in the four sellers in all contexts. D is, however, incomplete - A's trust for B in c_x is missing. As SVD does not compute on incomplete matrices, we insert the row's average to fill the gap. Given that the row's average is an arbitrary value, step 6 will not use its decomposition, as Troyanskaya *et al.* [22] suggested. For example, in Figure 1, TRULLO populates D and assigns the row's average 0.71 to the the value to be bootstrapped (underlined).

Step 4. Apply SVD on D thus obtaining $D = E \cdot F \cdot G^T$ (as Figure 1 shows).

Step 5. From the three resulting matrices, extract the elements F_{kk} and G^T_{kj}, $\forall k \in [1, m]$ and $j \in [1, n_c]$, where: F_{kk} is the kth feature's influence on trust assessment; G^T_{kj} is the kth feature's relevance in the jth context; and m is the number of relevant features[4]. In Figure 1, with $m = 3$, TRULLO extracts

---

[3] By definition, the number of feature is $n_f = min\{n_s, n_c\}$ [18]; however, in most applications $n_s > n_c$ and thus $n_f = n_c$.

[4] The number of features that the decomposition extracts from a matrix ($n_s \times n_c$) is $min\{n_s, n_c\}$. The number of contexts A knows is ($n_c - 1$). Therefore, the number m of relevant features is $min\{n_s, n_c, (n_c - 1)\}$, i.e., $min\{n_s, (n_c - 1)\}$.
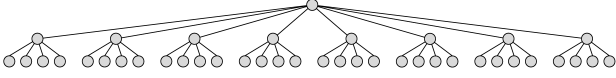
Fig. 2. Ontology of reference. We evaluate TRULLO by simulating its use in antique markets. The simulation models the number of contexts and their relationships with the above ontology. This reflects a typical ontology of the antique sections on eBay and Amazon.
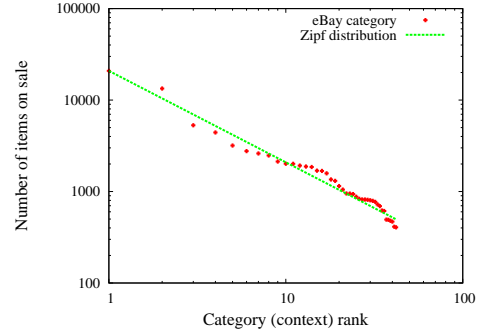

Fig. 3. Popularity of antique categories in eBay. The categories (contexts) are ranked by the number of antiques on sale. The resulting popularity follows a Zipf distribution.

the elements in bold from F and G^T, and also those underlined from the latter.

Step 6. For each lth context in which A has received ads from B (in our case, $l \in \{w, y, z\}$), regress A's trust in B (in the ith seller) in the lth context against the relevances in that context of all m features: $D_{il} = b_1 F_{11} G^T_{1l} + \ldots + b_m F_{mm} G^T_{ml}$. As we neither regress D_{ix} nor consider any feature in the bootstrapping context (G^T's column corresponding to c_x), the row's average in step 3 has little influence in the regression. That regression then results in m correlation coefficients $\{b_1, \ldots, b_m\}$. For example, in Figure 1, TRULLO regresses the elements in bold of D against those of F and G^T, and obtains the following correlation coefficients: $\{0.5886, -0.4578, 0.5734\}$.

Step 7. Having the correlation coefficients, the elements in bold of F, and those underlined of G^T, finally compute $t = D_{ix} = b_1 F_{11} G^T_{1x} + \ldots + b_m F_{mm} G^T_{mx}$. In words, Alice's trust in B in context c_x equals the *weighted* combination of Alice's trust values in B across features ($b_1, \ldots, b_k, \ldots, b_m$). The weighting factors for each kth feature are the feature's influence on trust assessment (F_{kk}) and the feature's relevance in the xth context (G^T_{kx}). In the example of Figure 1, TRULLO would set t to 0.8037.

That concludes the description of the bootstrapping steps. We now turn to evaluating TRULLO.

V. EFFECTIVENESS OF TRULLO: SIMULATION DESIGN

The goal of TRULLO is to *bootstrap* trust values on *hand-held* devices. To ascertain whether TRULLO meets this goal, our evaluation ought to answer the following two questions:

- *How effective is TRULLO in setting initial trust values?* This section describes the simulations we set up to answer this question; and Section VI analysis the results observed.
- *What computational overhead does our algorithm impose on a mobile phone?* Section VII describes the performance of a J2ME implementation of TRULLO on a Nokia mobile phone.

In our running example, only a subset of the sellers send ads that a visitor A finds relevant. A runs TRULLO to "guess" who those sellers are. As such, we measure the effectiveness of TRULLO as the extent to which the sellers selected by TRULLO actually send ads that A finds relevant.

Having a measure of effectiveness, we now need a system to test such effectiveness. The system we simulate consists of:

1) A set of contexts in which sellers publish and in which visitors receive advertisements.
2) For each seller in the simulation, a subset of the contexts of expertise is allocated to that seller so that the overall distribution of sellers across contexts is Zipf. Likewise, for each visitor, a subset of the contexts of interest are allocated to that visitor so that the overall distribution of visitors is Zipf.
3) In their contexts of interest, visitors select new sellers by bootstrapping trust, and calculate the utility of the ads provided by the selected sellers. TRULLO, plus two other popular bootstrapping techniques discussed in the literature, are used and compared.
4) Visitors rate ads based on their accuracy. The accuracy depends on the originating sellers' expertise. Each seller is deemed to be an expert in one context, but it also publishes ads in its other contexts, with an increasing likelihood of inaccuracy as the contexts of its ads "move away" from its area of expertise. This models sellers being not equally expert in all contexts.
5) Based on their ratings, visitors update their trust.

Before presenting the results of the simulations we ran, we now provide details about the setup of each of the five steps above.

1) Ontology Definition. Although we focus on evaluating the effectiveness of TRULLO at predicting trust *without* an ontology of contexts, it is necessary for us to craft one for the purposes of simulation. This ontology is used to model the characteristics of antique markets (i.e., number of contexts, and how visitors and sellers distribute their interests across them), but *not* to bootstrap trust. Looking at both Amazon and eBay, ontologies for antiques are flat, i.e., they have few levels and most of the nodes lie at lower levels. We hence consider an ontology of 40 contexts: 1 root, 8 children, each of which has 4 grandchildren (Figure 2).

4

2) Context Allocation. Having defined the ontology of contexts, we should now distribute sellers' expertise and visitors' interests across those contexts. Again, we looked at eBay. We have ordered the 42 lowest categories in the eBay antique section by the number of items on sale and have then plotted the result in Figure 3: a Zipf distribution best approximates the result; this is in line with literature expectation[5]. We assign to each context in the ontology previously defined a number of visitors and sellers such that the popularity of each context (ranked by the number of visitors and sellers) follows a Zipf distribution[6]. As a consequence, most visitors and sellers are associated with a few (popular) contexts and most ads will therefore be created for those contexts. We then consider that each seller is expert in one of the contexts with which it is associated (this context being chosen uniformly at random). Being expert in a context allows sellers to be able to produce highly-relevant ads in that context. Moreover, to account for real-life unpredictability in the behavior of sellers, we allow sellers to randomly change their expertise with a probability p (see Section VI).

3) Bootstrapping Methods. In their contexts of interest, visitors select the sellers from whom they want to receive ads. They do so by bootstrapping trust in sellers with TRULLO and with two alternative literature techniques: initial disposition bootstrapping [24] and recommendation-based bootstrapping [11]. We hence consider that each visitor bootstraps trust with three methods and records a sum of utility for each of them separately. Thus, in addition to TRULLO, we consider:

- *Initial disposition bootstrapping.* The visitor sets the initial trust to be a fixed value in a range $[0, 1]$ that reflects her disposition to trust. For example, that value might be 0.2 if she is pessimist, or 0.8 if she is optimist. In our simulation, a visitor setting the same initial trust (whatever that is) for all sellers is equivalent to a visitor randomly choosing a seller, thus making initial disposition bootstrapping the worst-case scenario.
- *Recommendation-based bootstrapping.* The visitor bootstraps trust in seller B in context c_x by

collecting recommendations from other visitors. We consider that recommenders send their actual trust values in B (i.e., they are wholly reliable) in context c_x (with no ontology misalignment). Under such assumptions, and given that sellers perform consistently over time, recommendation-based bootstrapping is the ideal case. To see why, consider that a recommendation about B is a record of how B performed. That record is objective in the sense that the recommender does not introduce any distortion (no ontology misalignment or fake recommendations). As B performs consistently over time, the recommendation is a predictor of how B will perform at all future points. Naturally, this is unrealistic but it serves as a yardstick (best case scenario) to evaluate TRULLO.

4) Ads Relevance. Visitors receive ads from the sellers they have selected. They then need to rate those ads but are unaware of the area of expertise of each seller. They simply obtain ads purporting to be within a particular context but, in reality, being of variable quality depending on the distance between the area of expertise of the seller and the context of the ad. Intuitively, the closer an ad to its seller's expertise, the more accurate and thus well-received the ad. This accounts for sellers not being equally expert in all contexts. Whenever it receives an ad, a visitor rates it as follows: $ad's\ relevance = w - dist(c_a, c_s)$, where c_a is the ad's context, c_s is the seller's context of expertise, and w is the maximum contextual distance (in our ontology $w = 4$). In words, the closer the ad's context to the seller's expertise, the higher the ad's relevance. In particular, as the distance between the two contexts increases, the relevance proportionally decreases from a maximum w (c_a and c_s are the same) to a minimum 0 (c_a and c_s are farthest). We compute the distance between c_a and c_s as the minimum number of edges between the two contexts[7].

5) Trust Updating. Upon rating seller B's ads, A's trust model updates its trust in B. To model this, it was necessary to integrate TRULLO with an available distributed trust model, and we elected to use B-trust [25], our Bayesian trust model for ubiquitous devices. The advantage of this model is that it formally updates trust according to Bayes' theorem on input of *discrete* ratings (not necessarily binary), and, having being designed for ubiquitous

---

[5]Zipf-like distributions are rooted in the dynamics of sending ads: the more ads a seller sends up to a certain moment, the more likely it is that the seller will send other ads in the future. This, which is a form of preferential attachment, is known to lead to power-law distributions, as shown in economics and complex networks [23].

[6]While there is good reason to suppose that the distributions of sellers to contexts, and visitors to contexts, are all Zipf-like (power-law with parameter *close* to unity), there is no real reason to suppose that they follow the 'strict' Zipf distribution (power-law with unitary parameter). However, slightly changing such a parameter (1 ± 0.2) demonstrated little effect on our simulation results; we thus report results for a unitary parameter.

[7]Of course, other measures of contextual distance might be defined. Indeed, we have also used a more complex distance function, whereby siblings of lower levels are considered closer than siblings at higher levels. This did not lead to statistically significant changes in the results.
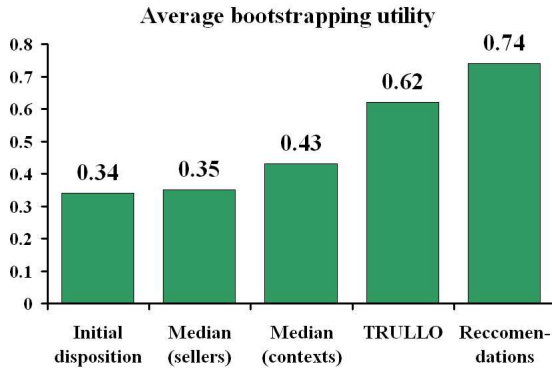
Average bootstrapping utility

Fig. 4. Average bootstrapping utility for TRULLO, two standard bootstrapping methods, and the median of trust values across sellers and across contexts. We average all utilities across all visitors, and each utility lies within $[0, 1]$. Recommendation-based bootstrapping relies on the ideal case of truthful recommenders (no fake recommendations) among which there is no ontology misalignment.
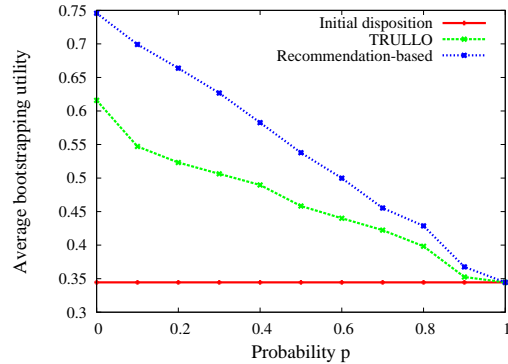

Fig. 5. Average bootstrapping utility for TRULLO and two standard bootstrapping methods as a function of the probability p of sellers randomly changing their expertise.

devices, it does so relying on a small data structure. Please refer to [25] for further details on B-trust and to [26] for one of its extensions.

Simulation Execution. We now describe how the five points above have been combined to perform a full simulation. To begin with, we have divided up the simulation execution into two phases:(1) visitors build initial knowledge, (2) upon which they then bootstrap trust.

- *Build initial knowledge.* Initially, sellers send ads of their items, and interested visitors receive them. To simulate this, in each round, each seller sends one ad in each of the contexts c_i with which it is associated. Visitors who are associated with c_i receive the ad and the process of rating and trust update proceeds as described above. Over the set of all sellers, most ads are produced in few contexts (those with most sellers) and few ads in most contexts (those with few sellers), simply as a consequence of allocating sellers to contexts according to a Zipf distribution.
- *Trust bootstrapping.* After the initial phase, whenever visitors wish to view ads in a given context, their devices have to choose the best seller(s) in that context. To model this, each visitor selects one new seller in each of the contexts c_j to which it has been assigned. To do so, for each context c_j, the visitor:
1. Bootstrap trust in c_j for all known sellers other than those we have already assigned to c_j;
2. Select the seller with the highest bootstrapped value;
3. Update its utility: utility_sum= utility_sum + $\frac{1}{w}(w - dist(c_j, c_s))$. In words, the closer c_j

to the seller's expertise (c_s), the higher the visitor's utility increase. To obtain an average utility within $[0, 1]$, we have normalized the contribution to the sum (multiplying it by $\frac{1}{w}$).

VI. EFFECTIVENESS OF TRULLO: SIMULATION RESULTS

We now discuss the results of a number of simulations we ran, based on the setup discussed above, to assess TRULLO's effectiveness in bootstrapping trust. We then analyze how some simulated factors might have affected the results.

TRULLO compared to two standard bootstrapping methods. We compare TRULLO to both initial disposition bootstrapping and recommendation-based bootstrapping. We simulate a typical antique market that consists of $N_s = 100$ sellers and of $N_v = 1000$ visitors. We run the first part of the simulation twice. As we will see, after two updates, trust values converge (i.e., their confidence is maximum) because sellers performs consistently over time. We then run the second part and average the utilities for each bootstrapping methods across all visitors. Figure 4 shows that TRULLO's average utility (0.62) is much closer to (the ideal) recommendation-based bootstrapping's (0.74) than to (the baseline) initial disposition bootstrapping's (0.34). Even if it relies on truthful recommendations with no ontology misalignments, recommendation-based bootstrapping does not reach the maximum average utility of 1 because some contexts have no specialized sellers (selecting which the resulting utility would be 1).

TRULLO compared to a simple median. One may now ask whether using a simple median instead of TRULLO would yield similar results. To see whether this is the case or not, consider a visitor bootstrapping its trust in seller S. We distinguish two cases:

- If the visitor knows k_s sellers in the bootstrapping context, it may set its initial trust as the
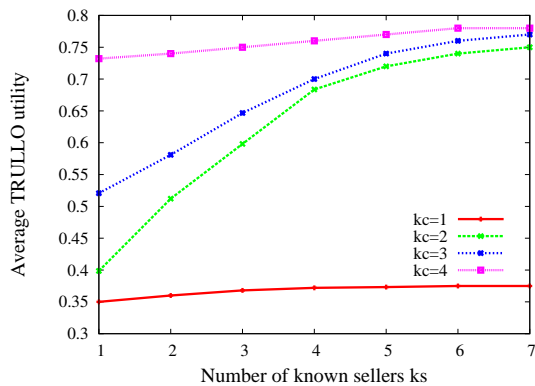
Fig. 6. Average bootstrapping utility for TRULLO versus its input size. Each utility refers to TRULLO bootstrapping trust in a context in which k_s sellers are known in other k_c contexts. We average across all visitors.

Fig. 7. Visitor's device performance (seconds) versus TRULLO input size, as means of 10 runs.

median of its trust values in those sellers (median across sellers).

- If the visitor knows S across k_c contexts, it may bootstrap trust as the median of its trust values in S across those contexts (median across contexts).

Figure 4 shows that TRULLO's average utility is much higher than that of the median (computed across either sellers or contexts). In particular, either median performs slightly better than initial disposition. Still, for any number of contexts/sellers, either median's utility is less than 0.45.

Factors affecting the effectiveness of TRULLO. Having these preliminary results, we now see how some simulated factors might have affected TRULLO's bootstrapping utility:

- *Confidence in the trust values upon which visitors bootstrap.* By decreasing the number of rounds of the first part of the simulation, visitors would receive fewer ads from sellers and thus run fewer trust updates, and they would then bootstrap trust upon more uncertain trust values. By doing so, TRULLO's average utility, however, does not significantly change because sellers perform consistently over time, and hence trust values in them converge just after receiving two ads from any of them in a given context. So, to capture real-life unpredictability in the behavior of sellers, we now allow sellers to send ads whose relevance reflects either their actual expertise with probability $(1 - p)$ or a randomly chosen expertise with probability p. Figure 5 shows that as p increases, the average utilities for TRULLO and recommendation-based bootstrapping decrease, as expected. However, even for high values of p, both bootstrapping methods perform better than initial disposition.
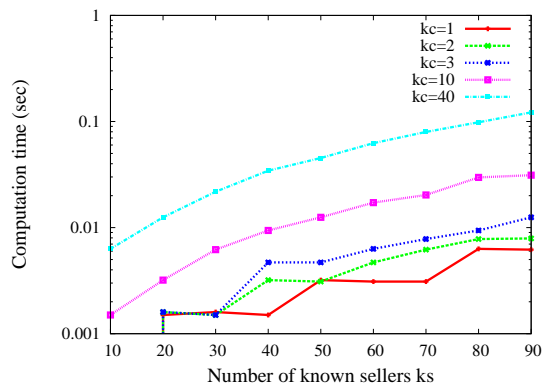
Also, their utilities decrease linearly. The reason is that for each 'misplaced' ad (i.e., ad that a seller sends from a context other than that of her expertise), the utility decreases by a certain amount on average. The number of 'misplaced' ads increases as p increases and, thus, the average utility *proportionally* decreases.

- *Size of the input matrix.* The size of TRULLO's input matrix depends on how sellers distribute across contexts. As the input size changes, bootstrapping accuracy might change, and that might affect the utilities of visitors using TRULLO. To see how, during the simulation execution, whenever TRULLO bootstrapped trust, we kept track of its utility and of the corresponding input size. We then averaged all utilities corresponding to the same size. Figure 6 shows that TRULLO performs better than initial disposition bootstrapping even on input of one single trust value. As one might expect, as the number of contexts and the number of sellers upon which TRULLO bootstraps increase, so does its average utility.

VII. COMPUTATIONAL COST OF TRULLO

Given that TRULLO effectively bootstraps trust, it is now worth checking whether it is usable on a mobile phone. To this end, we implemented TRULLO in J2ME and ran it on a Nokia 3230 mobile phone (whose features include: Symbian operating system 7.0, 32 MB of memory, 32-bit RISC CPU). Figure 7 shows TRULLO performance, given as the mean of 10 runs, for varying input $(k_s \times k_c)$ matrix sizes. We minimized background activities by shutting down all applications other than TRULLO. The computational overhead is very low. That is because TRULLO's input matrix only contains the ratings of a single user's past experiences. For example, to bootstrap a value in a context in which 20 sellers are known in 10 contexts (maximum input in the previous experiments), TRULLO takes just 3.2 milliseconds.

7

VIII. Discussion

Based on those evaluation results, we now discuss various open questions.

Correlation of how sellers perform across contexts. One inherent property of TRULLO design is that one can extract statistical features from trust values. Unfortunately, we do not have any trust set from real ubiquitous computing applications that corroborates this property. However, we do have Internet web sites reporting user ratings across contexts. Take hostels.com: it reports customer ratings of hostels across contexts, namely character, security, location, staff, fun, and cleanliness. By sampling parts of these trust ratings and applying the singular value decomposition, we learned that they do correlate across contexts - for example, trust ratings about 'character' and 'staff' roughly share the same statistical features. That, however, does not guarantee that this would be the case in all ubiquitous computing applications. For this reason, our simulation setup has made no explicit assumption on whether trust correlates across contexts, and it has shown how some simulated factors have affected the results.

Choosing the right bootstrapping method for a ubiquitous computing application. The choice of the right bootstrapping method is application-dependent. More concretely, consider the following aspects that are usually critical in ubiquitous computing:

- *Device computational cost.* If the computational cost must approach zero and bootstrapping accuracy does not matter, then initial disposition bootstrapping may be a fair choice. Otherwise, one might use TRULLO, which runs on a mobile at modest computational cost (Section VII).
- *Device communication overhead.* In a fully distributed setting, asking for recommendations might considerably increase data traffic among devices. To avoid that, one may use TRULLO since it is more effective than initial disposition bootstrapping and solely relies on local information (no device communication required).
- *Threats.* Modeling hostile environments is an important research question, on which we have not focused but now ponder briefly. Consider environments that may be deemed hostile because of either ontology misalignments among users, or presence of fake recommendations, or users behaving very differently across contexts. At the presence of the first two problems (which are indeed very likely), recommendation-based bootstrapping may suffer as it relies on third party information that, in this case, would be made unreliable by ontology misalignments and fake recommendations. In such a situation, one may prefer TRULLO as it relies only on local information. However, if sellers perform very differently across logically close/related contexts, then TRULLO may turn to be a poor choice as it no statistical property from sellers' behavior to extract would exist. Thus, at the presence of all three problems, one has to resort to initial disposition bootstrapping.

IX. Related Work

Literature includes three main ways to set initial trust. *First*, most of the reputation models in P2P networks [27], [28] and in social networks [29], [30] effectively bootstrap trust using recommendations, but they do so in a single context. It would be no trivial, yet interesting to extend these approaches using, for example, collaborative filtering techniques [31]. However, this extension needs further research for coping with fake recommendations and ontology misalignments among recommenders, and for scaling in a fully distributed setting of handheld devices.

The *second* proposition consists in assigning fixed trust bootstrapping values. Two examples include: Perich *et al.* [24], who define trust values based on the trustor's initial dispositions (pessimistic, optimistic, and undecided); and Buchegger *et al.* [10], who set the initial trust to a uniform distribution (as does our recent work [25]). Those choices apply only to specific problem domains (e.g., to packet forwarding).

The *third* proposition consists in setting the initial trust value for B in context c_x close to the trust values we already have about B in contexts *similar* to c_x. This may efficiently bootstrap trust, but, on the other hand, needs a measure of contextual distance to find out which are the contexts similar to c_x. Two recent types of approach might fill the gap: the first [12], [32] defines similarity between any two contexts in an ontology as the distance between the two corresponding nodes; the second type [33] draws context similarity based on a direct graph of contexts (a less-constrained structure than a tree) whose weights have to be, however, manually set by device users. The researchers who proposed the first type of approach have acknowledged that the idea of a universally accepted context ontology hardly belongs to reality and that additional techniques are needed (e.g., ontology matching for ubicomp [34]); those of the second concede that their solution has to be automated to be usable.

Compared to those three approaches, TRULLO is:

- *effective* - it sets initial trust value that are not arbitrary but are based on properties of its user's past experiences;
- *robust* to ontology misalignments among recommenders and to fake recommendations - it does not rely on collecting recommendations;
- *automated* - it does not need any user intervention.

X. CONCLUSION

We have shown that TRULLO effectively bootstraps trust by integrating it with B-trust (an existing trust model for ubiquitous devices) and by simulating its use in informal antique markets; in particular, TRULLO performs close to how exchanging recommendations would do in an ideal (though unrealistic) world, one in which recommenders are wholly truthful and, furthermore, share the same ontology. Our J2ME implementation of TRULLO does not impact the usability of a Nokia 3230 mobile phone.

In terms of ongoing work, we are currently studying the problem of how device A bootstraps its trust in B, if A does not know B at all.

REFERENCES

[1] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *Proc. of the 1st ACM Workshop on World Sensor Web (ACM Sensys)*, Boulder, USA, October 2006, pp. 117–134.

[2] H. Rheingold, *Smart Mobs: The Next Social Revolution*. Perseus Books Group, 2002.

[3] L. Kagal, T. Finin, and A. Joshi, "Trust-Based Security in Pervasive Computing Environments," *IEEE Computer*, December 2001.

[4] M. Sharmin, S. Ahmed, and S. I. Ahamed, "An Adaptive Lightweight Trust Reliant Secure Resource Discovery for Pervasive Computing Environments," in *Proc. of the 4th IEEE International Conference on Pervasive Computer and Communications (PerCom)*, Pisa, Italy, March 2006, pp. 258–263.

[5] L. McNamara, C. Mascolo, and L. Capra, "Trust and Mobility Aware Service Provision for Pervasive Computing," in *Proc. of the 1st International Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI)*. Springer, May 2006, pp. 603–610.

[6] M. Deriaz and J.-M. Seigneur, "Trust and Security in Spatial Messaging: FoxyTag, the Speed Camera Case Study," in *Proc. of the 3rd International Conference on Privacy, Security and Trust (PST)*, Ontario, Canada, October 2006.

[7] L. Capra, "Engineering human trust in mobile system collaborations," in *Proc. of the 12th International Symposium on Foundations of Software Engineering*. Newport Beach, USA: ACM Press, November 2004, pp. 107–116.

[8] V. Cahill, E. Gray, J.-M. Seigneur, C. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, G. Serugendo, C. Bryce, M. Carbone, K. Krukow, and M. Nielsen, "Using Trust for Secure Collaboration in Uncertain Environments," *IEEE Pervasive Computing Mobile and Ubiquitous Computing*, vol. 2, pp. 52–61, August 2003.

[9] S. Toivonen, G. Lenzini, and I. Uusitalo, "Context-aware Trustworthiness Evaluation with Indirect Knowledge," in *Proc. of the 2nd International Semantic Web Policy Workshop (SWPW)*, Athens, USA, November 2006.

[10] S. Buchegger and J.-Y. L. Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," in *Proc. of the 2nd Workshop on the Economics of Peer-to-Peer Systems*, Cambridge, USA, June 2004.

[11] A. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities," in *Proc. of the 33rd IEEE Hawaii International Conference on System Sciences*, vol. 6, Washington, USA, January 2000, p. 6007.

[12] J. Liu and V. Issarny, "Enhanced Reputation Mechanism for Mobile Ad Hoc Networks," in *Proc. of the 2nd International Conference on Trust Management*, vol. 2995. Oxford, UK: LNCS, March 2004, pp. 48–62.

[13] H. Abdi, *Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD)*. Encyclopedia of Measurement and Statistics. Thousand Oaks (CA), 2007.

[14] J. Coutaz, J. Crowley, S. Dobson, and D. Garlan, "Context is key," *Communications of the ACM*, vol. 48, no. 3, pp. 49–53, 2005.

[15] J. R. Douceur, "The Sybil Attack," in *Proc. of the 1st International Workshop on Peer-to-Peer Systems*, Cambridge, USA, 2002, pp. 251–260.

[16] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "SybilGuard: defending against sybil attacks via social networks," in *Proc. of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Pisa, Italy, May 2006, pp. 267–278.

[17] D. Quercia, S. Hailes, and L. Capra, "TATA: Towards Anonymous Trusted Authentication," in *Proceedings of the 4th International Conference on Trust Management*. Pisa, Italy: LNCS, May 2006, pp. 298–312.

[18] G. Forsythe, M. Malcom, and C. Moler, *Computer Methods for Mathematical Computations*. Prentice Hall, 1977.

[19] D. Quercia and S. Hailes, "Risk Aware Decision Framework for Trusted Mobile Interactions," in *Proc. of the 1st IEEE/CreateNet International Workshop on The Value of Security through Collaboration*, Athens, Greece, September 2005.

[20] A. Serjantov and R. Anderson, "On dealing with adversaries fairly," in *Proc. of the 3rd Annual Workshop on Economics and Information Security*, Minnesota, USA, May 2004.

[21] R. Hardin, "The street-level epistemology of trust," *Politics and Society*, vol. 21, no. 4, pp. 505–529, December 1993.

[22] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, June 2001.

[23] A. L. Barabasi, *Linked: How Everything Is Connected to Everything Else and What It Means*. Penguin, 2003.

[24] F. Perich, J. Undercoffer, L. Kagal, A. Joshi, T. Finin, and Y. Yesha, "In Reputation We Believe: Query Processing in Mobile Ad-Hoc Networks," in *Proc. of the 1st IEEE International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, USA, August 2004, pp. 326–334.

[25] D. Quercia, S. Hailes, and L. Capra, "B-trust: Bayesian Trust Framework for Pervasive Computing," in *Proceedings of the 4th International Conference on Trust Management*. Pisa, Italy: LNCS, May 2006, pp. 298–312.

[26] A. Mitra and M. Maheswaran, "Benefits of Targeting in Trusted Gossiping for Peer-to-Peer Information Sharing," in *Proc. of the IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Long Beach, USA, March 2007.

[27] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *Proc. of the 10th ACM International conference on Information and Knowledge Management*, Atlanta, USA, November 2001, pp. 310–317.

[28] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," in *Proc. of the 12th ACM International Conference on World Wide Web (WWW)*, Budapest, Hungary, May 2003, pp. 640–651.

[29] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proceedings of the 13th International Conference on World Wide Web*, New York, USA, May 2004, pp. 403–412.

[30] J. Sabater and C. Sierra, "Reputation and social network analysis in multi-agent systems," in *Proc. of the 1st ACM International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Bologna, Italy, June 2002, pp. 475–482.

[31] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: Applying collaborative filtering to Usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[32] L. Capra, "Reasoning about Trust Groups to Coordinate Mobile Ad-Hoc Systems," in *Proc. of the 1st IEEE Workshop on the Value of Security Through Collaboration*, Athens, Greece, September 2005.

[33] M. Kinateder and K. Rothermel, "Architecture and Algorithms for a Distributed Reputation System," in *Proc. of the 1st International Conference on Trust Management*. Crete, Greece: LNCS, May 2003, pp. 48–62.

[34] L. Capra, "MaLM: Machine Learning Middleware to Tackle Ontology Heterogeneity," in *Proc. of the 4th IEEE Workshop on Middleware Support for Pervasive Computing (PerWare)*, New York, USA, March 2007.