Tag Recommendations for SensorFolkSonomies

Juergen Mueller
L3S Research Center
University of Kassel
Wilhelmshöher Allee 73
Kassel, Germany
mueller@cs.uni-kassel.de

Stephan Doerfel
University of Kassel
Wilhelmshöher Allee 73
Kassel, Germany
doerfel@cs.uni-kassel.de

Martin Becker
L3S Research Center
University of Würzburg
Am Hubland
Würzburg, Germany
becker@informatik.uni-wuerzburg.de

Andreas Hotho
L3S Research Center
University of Würzburg
Am Hubland
Würzburg, Germany
hotho@informatik.uni-wuerzburg.de

Gerd Stumme
L3S Research Center
University of Kassel
Wilhelmshöher Allee 73
Kassel, Germany
stumme@cs.uni-kassel.de

ABSTRACT

With the rising popularity of smart mobile devices, sensor data-based applications have become more and more popular. Their users record data during their daily routine or specifically for certain events. The application WideNoise Plus allows users to record sound samples and to annotate them with perceptions and tags. The app documents and maps the soundscape all over the world. The procedure of recording and including the assignment of tags, has to be as easy-to-use as possible. We therefore discuss the application of tag recommender algorithms in this particular scenario. We show, that this task is fundamentally different from the well-known tag recommendation problem in folksonomies as users do no longer tag fix resources but sensory data and impressions. The scenario requires efficient recommender algorithms that are able to run on a mobile device alone, since Internet connectivity is not always available. Therefore, we evaluate the performance of ten tag recommendation algorithms and discuss their applicability in the mobile sensing use-case.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Tag recommendation, cold-start problem, SensorFolkSonomy, item features, mobile sensing, citizen science

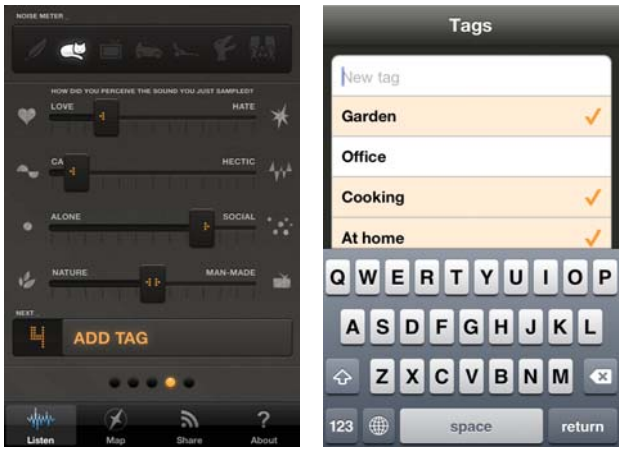
1. INTRODUCTION

Mobile devices like smartphones and tablets have become widely used and are still increasing in popularity. Their embedded sensors like GPS, microphone, accelerometer, or gyroscope enable multiple sensing applications [13]. Among the measurable quantities are environmental conditions like location, acceleration, orientation, or noise level. Users can complement this objective (i.e., measurement) data by subjective impressions using the user interface of an application. [6] suggest that there is a wide range of applications for mobile sensing and expect a rapidly growing field for urban sensing.

Citizen science is such a field of urban sensing where a large numbers of individuals contribute with small amounts of information to a larger dataset to be analyzed by researchers. Their goal is often to take part in the broadening of knowledge about ourselves and our environment [9]. The EU research project EveryAware[1] is one example of such initiatives. The project's goal is to cause a change in people's awareness towards their environment through insights about their soundscape [2, 3]. One of our most active user groups measure the noise pollution around the Heathrow airport in London to draw attention to their interests.

To obtain this goal, the EveryAware team offers the smartphone application WideNoise Plus – an application to measure and annotate samples from the soundscape using the build-in microphone. Its users can choose to record the average noise level in decibels (dB) over a time span of 5, 10, or 15 seconds. Further, they can state their perception about the recorded noise using four sliders (see Figure 1(a)). There is a slider to express whether they love or hate the noise, whether they perceive it as calm or hectic, whether they are alone or in a social situation, and whether the noise was natural or man-made. Finally, users can add keywords (i.e.,

[1] http://everyaware.eu/



(a) Perceptions view. (b) Tagging view.

Figure 1: Screenshots of WideNoise Plus for iOS.

tags) to the record to describe it further (see Figure 1(b)). WideNoise Plus was originally developed by WideTag Inc. and was then transferred to EveryAware in 2011 and is available for iOS[2] and Android[3].

The workflow of taking measurements, choosing perception, and annotating tags must be as easy as possible, since often users are not experienced with mobile technologies or would be discouraged by too complex or time consuming procedures. We therefore deal with the task of recommending tags for records in order to assist users in selecting appropriate keywords.

This task of suggesting such tags is related to the well-known tag recommendation problem from tagging systems (i.e., folksonomies) [11]: Suggest some tags to a user for a given resource. However, the tag recommendation in our case differs fundamentally from that in folksonomies by its resources, i.e., sensor data. While tag recommendation by itself is well studied, to the best of our knowledge, there are no studies about the tagging behavior when the tagged resources are sensory information. A second aspect that distinguishes the classical scenario from ours is that the recommendations have to be computed on mobile devices, whereas web tagging systems typically rely on strong computers and large data sets with precomputed statistics. Heavy computations would drain the battery too much, which could encourage the user to remove our application from their phone. As an urban sensing application, the application cannot rely on online resources, as is it supposed to be used anywhere, regardless of Internet connectivity. The annotated WideNoise Plus records are sent to the EveryAware server later as soon as reception is available.

The contributions of the paper are: A formal definition of the tag recommendation problem for sensor data, the evaluation of ten recommender algorithms using real-world data from WideNoise Plus, and the discussion of their performance regarding their resource consumption and their suitability for mobile devices.

The remainder is structured as follows: We will give a short overview on the related work in the following Section 2. In Section 3 we formalize the task and establish the difference to classical tag recommendations. Further, we introduce proximity-, and perception-based strategies to produce tag recommendations. Next, we describe the experimental setup, including the dataset description in Section 4. The experiments' results are presented and discussed in Section 5. Finally, we will conclude the paper and point to several aspects for future work.

2. RELATED WORK

We will cover two areas of related work, which are relevant for our scenario. First we will introduce the current state of tag recommendation research and how it is applicable to our scenario. Then we will cover the specifics of urban sensing and discuss the motivation for users to tag.

The task of recommending suitable tags for a given resource to a specific user has been subject to many studies. A broad variety of solutions has been proposed so far. [12] compares eight algorithms like popularity-based, collaborative filtering using similarities-based on both user-tag and user-resource vector spaces, and graph-based approaches like FolkRank. More recent work focusses on hybridization [5][7] of algorithms or advanced technology like tensor factorization models [15].

To our scenario only the popularity-based algorithms are directly applicable as all others depend on a set of fixed resources. Ubiquitous data as defined by [10] has some specific characteristics that makes it different from conventional data. Usually, many different types of data are involved and data emerges from a high number of partially overlapping, loosely connected sources. An important characteristic is that it is annotated with timestamps and geographic location. Usually, resource do not exists a second time as only rarely the same measurement is taken at the same place and perceived in the same way. This makes ubiquitous data different from traditional resources in recommender systems. Therefore, graph or tensor-based models would have to be modified to a great extent in order to fit our scenario. In the following section we will go into more detail about how the nature of sensory data is different from that typically found in tagging systems.

The presence of location information enables the use of spatial approaches. [1] presents a cluster-based tag recommendation approach for images from the social tagging platform Flickr[4]. The approach applies the k-Means clustering algorithm [16, page 62–63] on the location information and has been evaluated on the CoPhIR dataset [4]. In the latter work, the influence of geographical coordinates, low-level image features (e.g., color layout), and tag similarities on recommendations were compared. The results of their studies show that the geographical coordinates are the most helpful image descriptors. The approach is further described in Section 3.2.5 and will be examined in our evaluation.

Finally, [8] deals with the way people tag their resources, which is deeply related to the reasons why they tag. The most important reasons that can be applied to the WideNoise Plus scenario are "contribution and sharing", "attract attention", and "opinion expression". This becomes evident while looking at tags that were used during an EveryAware campaign taking place around the London Heathrow Air-

[2] https://itunes.apple.com/app/id657693514

[3] https://play.google.com/store/apps/details?id=eu.everyaware.widenoise.android

[4] http://flickr.com/

port. With their tags, the participants expressed and shared their frustration about the noise pollution caused by the airport.

3. TAG RECOMMENDATION

As outlined in the introduction, there are some special conditions for mobile sensing that must be addressed when choosing recommender algorithms. WideNoise Plus is most often used outdoors without regard to Internet connectivity. Thus, the application must be able to produce recommendations only from data that has been stored on the device and the elements of the current record (the measured noise, the location, and the user's perceptions). Furthermore, producing recommendations should only consume as little power and runtime as possible. Otherwise, the increased battery drain and long waiting time would discourage users from taking further measurement.

At the moment, it theoretically is still possible to store all relevant tagging information of all users of WideNoise Plus (approximately 150 KB) within the app on a smartphone. However, since the dataset is growing, algorithms that need only some (externally precomputed) subset or aggregation of the historic data are preferable to those that require the complete unprocessed data as their input.

In the following, we first formalize the tag recommendation task for the combination of sensor measurements and subjective perceptions. We then present each of the algorithms, which we will later use and compare.

3.1 Task Definition

Our tag recommendation task is closely related to that of tagging systems where users assign tags to resources like videos, photos, websites, or papers. Each of such systems underlies a structure called a folksonomy. It can be formalized as a quadruple $\mathsf{F} = (U, T, R, Y)$ [11] of the sets U containing all users, R containing all resources, and T containing all tags of the system. Finally, $Y \subseteq U \times T \times R$ is a ternary relation containing a triple (u, t, r) if the user $u \in U$ has assigned the tag $t \in T$ to the resource $r \in R$. Typical for a folksonomy is that the same resource is tagged by several users and that the same tag is used for several resources. Thus, the tag assignments in Y form connections among the entities of F. The task of recommending tags is then to suggest tags, given a user and a resource.

At first glance, the WideNoise Plus setting fits exactly this problem description. We can identify users and tags and use the different measurements as resources. However, the description does not cover the complexity of our scenario. Each record contains several attributes like the noise level, the location, a timestamp, and the perceptions entered by the user. While certainly in the tagging systems mentioned above the resources come with different attributes as well, in those cases – unlike the perceptions – the properties (e.g., the title of a paper, the content of a video) can be derived from the resource alone. They are the same for any user who has tagged the resource and they are (with the exception of updates) stable. It is therefore desirable, to exclude the perceptions from a resource and to model them as extra entities. Further, the sharing of resources can hardly be modelled in the same way as in folksonomies. Where in tagging systems, one of the key ideas is that users can see, discover, and copy other users' resources, this is not the same for the WideNoise Plus scenario. While users can tag several noise levels at the same location and time, they do not share the same resources (i.e., measurements). Rather, each measurement is an individual resource and (within the boundaries of physics) almost any combination of sensor measurements is conceivable. For example on the (approximately) same location, the noise measurements of two users might differ due to different times of day or week, different seasons, different company, or events. Moreover, slight variations in the location (that must not even be measurable with GPS) can yield significantly different noise measurements, e.g., being inside a building or outside right next to it. Despite these difficulties, the properties and the perceptions are surely reasonable candidates for exploitation in recommendations. Therefore, we modify the above described notion of folksonomy to that of SensorFolkSonomy as follows.

DEFINITION 1. *Given a set of users U, a set of tags T, a set S of sensors with ranges $S_i, i = 1, ..., |\mathsf{S}|$, and a set P of perception categories with ranges $P_j, j = 1, ..., |\mathsf{P}|$ a SensorFolkSonomy is the 5-tuple S defined as*

$$\mathsf{S} := (U, T, S, P, Y)$$

Hereby, $S = \prod_{i=1}^{|\mathsf{S}|} S_i$ and $P = \prod_{j=1}^{|\mathsf{P}|} P_j$ so that $Y \subseteq U \times T \times S \times P$ is the tag assignment relation. A record is a tuple $(u, T_{(u,s,p)}, s, p)$, where $u \in U$, $s \in S$, $p \in P$ and $T_{(u,s,p)} := \{t \in T \mid (u, s, p, t) \in Y\}$, such that $T_{(u,s,p)} \neq \emptyset$.

In the definition, the symbol $\prod$ as usual denotes the Cartesian product, S is the space of all possible measurement combinations and P is the space of all possible perception combinations. The resulting records are similar to posts in regular folksonomies. The tag recommendation task for a SensorFolkSonomy is now to suggest tags given a user u, a vector of measurements s and a vector p of statements in each perception category.

According to our definition, a WideNoise Plus record can be modelled in the following way: It is recorded by a user $u \in U$ who is identified by the unique ID of their mobile device. The unique ID of the mobile device is part of the record and we assume that different devices usually imply different users. The content of one record is given by a vector $s \in S$ holding the measured noise level as well as the coordinates of the location. The user can annotate this reading with a freely chosen set of tags $t \in T$ and choose values in 4 perception categories: feeling, disturbance, isolation, and artificiality (thus $|P| = 4$).

The task of tag recommendation is to recommend, for a given user $u \in U$, a given tuple of sensor measurements $s \in S$, and a given tuple of perceptions $p \in P$, a set $\tilde{T}_{(u,s,p)} \subseteq T$ of tags. In many cases, $\tilde{T}_{(u,s,p)}$ is computed by first generating a ranking on the set of tags according to some quality or relevance criterion, from which then the top k elements are selected.

3.2 Recommendation Approaches

We compare several approaches against each other in our experiments. We describe them in the remainder of this section and discuss their advantages and drawbacks regarding resource consumption as well as their suitability for the mobile environments.

3.2.1 Most Popular Tags (MPT)

A very simple recommendation method is to always suggest those tags $t \in T$ that have been assigned the most often

so far, i.e., where the set $T_t := \{(u, t, s, p) \in Y \mid u \in U, s \in S, p \in P\}$ is largest. This yields a non-personalized recommender that will serve as a lower baseline in our comparison of algorithms.

The only input data that would have to be provided for the app are just those top most popular tags. Since also nothing has to be computed, the application would require only very little storage and almost no processing time at all. Therefore, this method would be the best-case in terms of resource requirements.

However, it is expected to be rather bad with regard to the quality of the recommendations, since it is just a static list of the same tags for each record. Table 1 shows the list of the current most popular tags. While there are some country specific tags like the Italian word "esterno" (outdoor scene), there are some international ones like "garden" or "car" that are likely to occur all over the world. Therefore, this recommendation strategy is considered an adequate baseline for our evaluations.

Table 1: The 10 most popular tags in the dataset

Amount	Tag
573	garden
557	esterno
549	heathrow
525	aeroplane noise
271	voci
187	car
181	antwerpen
157	plane
151	street
133	arriva

3.2.2 Most Popular Tags by User (MPTU)

Another very simple recommendation method is to suggest those tags $t \in T$ that have been used by the given user $u \in U$ the most often so far, i.e. where the set $T_{u,t} := \{(u, t, s, p) \mid s \in S, p \in P\}$ is largest. This yields a personalized recommender that recommends tags that are known to the user and in a language they understand.

It is also very suitable for the mobile devices, as only the user profile and no other training data has to be stored. Using the pre-ordered list of the user's tags, the algorithm is similarly fast as the global most popular tag recommender. However, this algorithm has a severe cold start problem as it cannot produce tags for new users.

3.2.3 Proximity-Based Approach (Prox)

An approach that uses the location information provided by the location sensor $s \in S$ is to recommend tags that have been used so far at the given location or nearby. Prox is thus a context-aware recommender that will recommend tags that likely describe the location like for example "airoplane noise", which has been used near airports. Therefore, a proximity-based prediction is likely to have good performance.

The algorithm has stronger requirements than the previous ones. Either the whole dataset (all recordings in any location) must be stored on the device or an Internet connection is required beforehand in order to query for records that have been taken roughly near the user's current location.

In our experiments we will use the k-Nearest-Neighbors algorithm [16, page 129–131] to find the nearby tags. This ensures that this approach always recommends tags even if they are taken from faraway places. For our experiments we manually choose a value of 42 for k, since this showed good results in a subset in the training data.

The distance between two locations can be calculated with a number of methods like the Manhattan, the Euclidean, or the great circle distance. The Manhattan distance is particularly is rather inaccurate although very easy to compute. The Euclidean distance is much better in terms of accuracy, but with the price of a higher computational effort. However, compared to the actual air-line distance, the accuracy is getting worse for locations further away from the equator. Finally, the great circle distance is very precise, but is the most expensive with regard to computations. We use the Euclidean distance (Prox-ED) and the great circle distance (Prox-GCD) due to its higher accuracy.

3.2.4 Perception-Based Approach (Perc)

An approach that uses WideNoise Plus's perception values $p \in P$ is able to recommend tags $t \in T$ that are associated with the same mood (e.g., "love"). This yields a context-aware recommender that will recommend tags that describe the user's perception of the noise, location, etc. (e.g., "noisy plane spoiling peace"). There are four scales with a range from -5 to +5 each with steps of size 1 to express the corresponding perception (see Figure 1(a)):

- Feeling: Ranges from "hate" to "love" and expresses whether the user enjoys the recorded noise or whether it was unpleasant.
- Disturbance: Ranges from "hectic" to "calm" and expresses how disturbing the recorded noise was perceived by the user.
- Isolation: Ranges from "alone" to "social" and expresses how much company the user had.
- Artificiality: Ranges from "man-made" to "nature" and expresses whether the recorded noise was caused by humans, machines, or nature.

The method is suitable for mobile devices, as only an aggregated list of tags for each possible perception combination has to be stored. Using the pre-ordered lists of the perception's tags, the algorithm has to combine those lists that are the most similar to the given perception setting. A perception vector p is considered similar to the current perception vector p if no perceptions differ more than a given threshold d, i.e. if $||p - p||_\infty \leq d$. In our experiments we will set the (initial) threshold to $d = 1$ and increase it by one in cases where no such measurement p exists and thus nothing could be recommended.

3.2.5 Clustering-Based Most Popular (Clus)

This approach, presented by [1] uses the location information of the location sensor to cluster the records and assign the most frequent tags $\in T$ ordered by decreasing user frequency of a cluster's records to that cluster during a preprocessing step. Recommended are those tags that have been used in the cluster of a given location so far, i.e., those tags $t \in T$ where the set $\{(u, t, s\ , p) \mid u \in U, s\ \in C(s), p \in P\}$

is large. Hereby $C(s)$ is the cluster that the current location s belongs to. This yields a context-aware recommender that will likely recommend tags that describe the location. This algorithm is similar to Prox, but, since the records are clustered, the computational effort and the amount of input data is lower.

It is thus suitable for mobile devices, as only the precomputed ranked list of tags of each cluster have to be stored. For each new record, the distance to all clusters has to be computed to select the ranked tag list of the cluster closest to the user.

In an offline preprocessing, the resources are clustered using k-Means and the most frequent tags for each cluster are determined. k-Means requires the number of cluster k as an input parameter as well as a distance computation function. For k we use the rule of thumb proposed by [14, page 365]:

$$k \approx \left(\frac{n}{2}\right)^{\frac{1}{2}}$$

Hereby, n refers to the number of resources to be clustered and the Euclidean distance is used as distance function. Clusters are represented by their centroids and in the recommendation phase, we use the Euclidean distance for distance calculation.

During our experiments we discovered that, in our scenario, it is better to choose the absolute tag frequency during clustering phase rather than the user frequency. We will present the result for user frequency (i.e., Clus-UF) and absolute tag frequency (i.e., Clus-AF) separately during our evaluation.

3.2.6 *Hybridization*

To improve performance, multiple recommenders can be combined in hybrid recommenders. Such a combination can improve the results by combining several aspects, e.g., to yield a location-based approach that also is influenced by the given perceptions.

The suitability for our scenario depends on the algorithms that are combined. In this paper we will analyze 3 combinations between most popular tag by user on the one hand and either the perception (Perc-MPTU), proximity (Prox-ED-MPTU), or clustering (Clus-ED-MPTU) approach on the other hand. We use most popular tag by user as it produces personalized recommendations with only little computational effort. In order to keep the computational effort small we chose the Euclidean distance-based versions of Perc and Prox.

All involved algorithms compute their individual rankings. For a tag $t \in T$ we compute a score as an unweighted linear combination [5] of the inverse ranks according to the following equation:

$$score(t) = \left(\frac{1}{\text{rank}_1(t)} + \frac{1}{\text{rank}_2(t)}\right)^{-1}$$

Hereby, $\text{rank}_1(t)$ and $\text{rank}_2(t)$ are the positions of the tag t in the rankings of the two combined algorithms.

4. DATASET AND EXPERIMENTS

In this section we introduce the dataset of our analysis and how it was assembled as well as the metrics we use for the evaluation in Section 5.

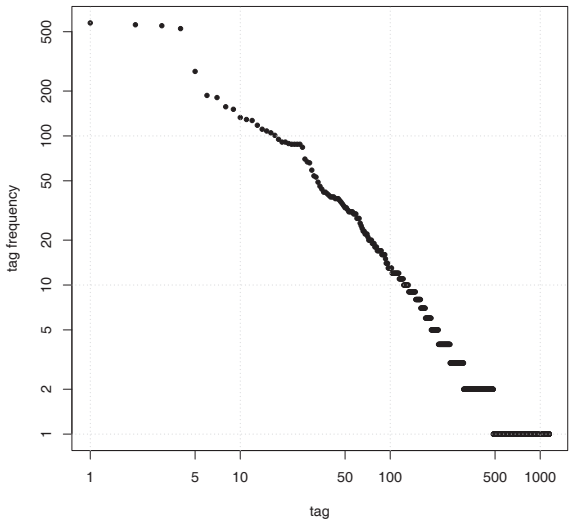


Figure 2: Distribution of the tag frequency on a log-log scale. The elements on the x-axis are the 1,151 unique tags, ordered by decreasing frequency.

4.1 Dataset

The basis for our experiments is the full set of WideNoise Plus records with at least one tag, collected between December 14, 2011 and June 12, 2013. After the removal of records that had been submitted for testing by the developers, the collection consists of 5,434 reports collected by 546 users that contain 1,151 distinct tags and 9,255 tags in total. The following further preprocessing steps were applied to the tags: All tags have been lower-cased and some encoding issues have been resolved manually (e.g., we replaced "wrzburg" with "würzburg").

Before we describe the experiments on tag recommendations, we observe a few statistical properties of the datasets. Figure 2 shows the distribution of the tag frequency. The distribution tends to be fat tailed.

Figure 3 shows the distribution of the number of tags assigned to one record. The maximum number of assigned tags is 8 and we therefore pick it as the maximum number of recommended tags in our experiments. On average, one WideNoise Plus record has 2.45 tags assigned to it.

Figure 4 shows the distribution of the number of tag assignments per user. The most active user assigned 2,461 tags and the average number of tag assignments per user is 33.92. However, we have a fat tail of users that made just one tagged records and then stopped using this feature.

4.2 Evaluation

We evaluate the different recommendation algorithms in an offline experiment. We split the full dataset into training and test data using a time split after 70 % of the records leaving 3,805 records for the train phase and 1,629 records for the evaluation phase. In that way, we stay close to the actual scenario: The WideNoise Plus app runs on a mobile device and must produce recommendations from the data on the device. While it is not possible to send training data

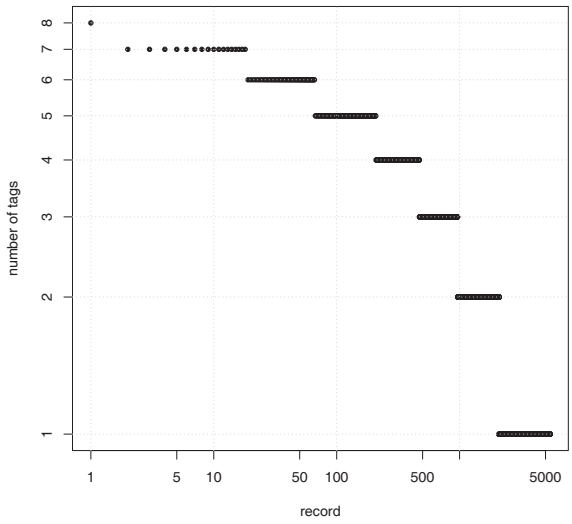


Figure 3: Distribution of tags per record on a log-log scale. The x-axis represents the dedicated records and the y-axis represents the number of tags assigned to such a record.

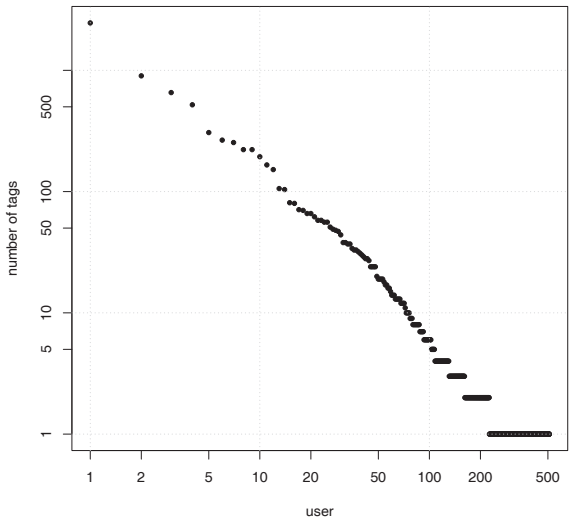


Figure 4: Distribution of the number of tag assignments per user. The x-axis represents the users and the y-axis represents the number of tag assignments of these users.

record by record to an application, it is very well conceivable, to update the app with training data in larger regular intervals. A consequence of this procedure is that the test data set contains users and tags that do not occur in the training data. Again, this is close to the real scenario, where often users take measurements over only a short time span and thus do not have large user profiles to be used for training. This closeness to the real-world scenario was the decisive element for a time split and against other methods like cross validation procedures, where random samples of the full data set are selected as test data.

The algorithms are trained and then used to produce a ranked list of recommendations $\tilde{T}_{(u,s,p)}$ for each record in the test dataset comprising the user u, the sensor measurements s (longitude, latitude and noise level) and the four perceptions p. To evaluate the performance we measure the predictive power of recommendations, i.e., for every record of the test data, precision, recall, and F_1 measure are computed. For these three metrics, the number of recommended tags has to be set to some fix number k. To pay tribute to the size of mobile devices and following the findings above on the maximum number of assigned tags, we let k run from 1 through 8 and compute the score at each level. Thus, if $T_{(u,s,p)}$ is the set of tags that were actually assigned to the record and $\tilde{T}_{(u,s,p)}$ is the set of the top k recommended tags, then precision and recall are defined as follows [16, page 109]:

$$\text{Precision}(u,s,p) = \frac{|\tilde{T}_{(u,s,p)} \cap T_{(u,s,p)}|}{|\tilde{T}_{(u,s,p)}|}$$

$$\text{Recall}(u,s,p) = \frac{|\tilde{T}_{(u,s,p)} \cap T_{(u,s,p)}|}{|T_{(u,s,p)}|}$$

The F_1 measure is the harmonic mean of precision and recall:

$$\text{F}_1(u,s,p) = 2 \cdot \frac{\text{Precision}(u,s,p) \cdot \text{Recall}(u,s,p)}{\text{Precision}(u,s,p) + \text{Recall}(u,s,p)}$$

Theoretical upper bound.

In the experiments, we will compare not only different algorithms against each other, but also to a theoretical "perfect recommender". This upper bound demonstrates, how much room for improvements is left for further, possibly more advances methods in future work. The bound is constructed by recommending those tags for a record that have actually been used for it as long as these tags occur in the training data. It is clear that no real algorithm – which of cause has no knowledge about the user's actually chosen tags – can beat that upper bound (unless it can produce tags that have never been used before).

5. RESULTS

In the evaluation we use the global most popular recommender as baseline. Every more sophisticated recommender should achieve better results than that. Additionally, we include the values for a perfect pseudo-recommender that predicts just those tags that are actually used and present in the train dataset. As described in Section 3.2.1, there are language specific tags in the list of the global popular tags (see Table 1). Further, besides the larger geographic areas, there are also tags that will occur only in certain particular areas like the tag "heathrow", which would be relevant only around London. It is therefore to be expected, that the most popular baseline will achieve rather low results.

Figure 5 shows the results for precision and recall. The F_1 measure results are shown in more detail in Figure 6 and Table 2. In the discussion, we focus on the scores that

Table 2: F_1 measure for WideNoise Plus

number of tags	MPT	MPTU	Perc	Prox-ED	Prox-GCD	Clus-UF	Clus-AF	Perc-MPTU	Prox-ED-MPTU	Clus-AF-MPTU	upper bound
1	0.002	0.180	0.084	0.213	0.204	0.049	0.165	0.165	0.234	0.186	0.433
2	0.028	0.175	0.113	0.203	0.204	0.054	0.192	0.204	0.227	0.229	0.450
3	0.043	0.162	0.100	0.181	0.182	0.047	0.178	0.193	0.206	0.204	0.384
4	0.036	0.153	0.098	0.170	0.169	0.064	0.164	0.165	0.190	0.188	0.327
5	0.032	0.149	0.091	0.153	0.162	0.057	0.155	0.146	0.168	0.176	0.284
6	0.030	0.135	0.084	0.148	0.146	0.051	0.138	0.139	0.163	0.161	0.251
7	0.027	0.124	0.082	0.136	0.134	0.046	0.128	0.132	0.149	0.149	0.225
8	0.026	0.116	0.076	0.126	0.125	0.042	0.120	0.125	0.139	0.139	0.204

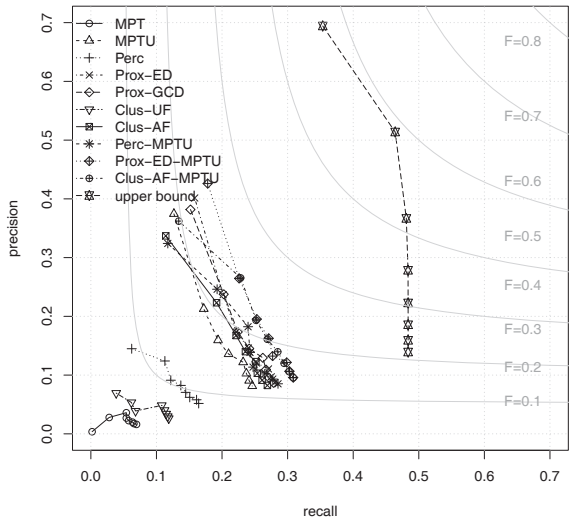


Figure 5: Recall vs. precision for WideNoise Plus. The grey lines are isoquants for several F_1 scores.

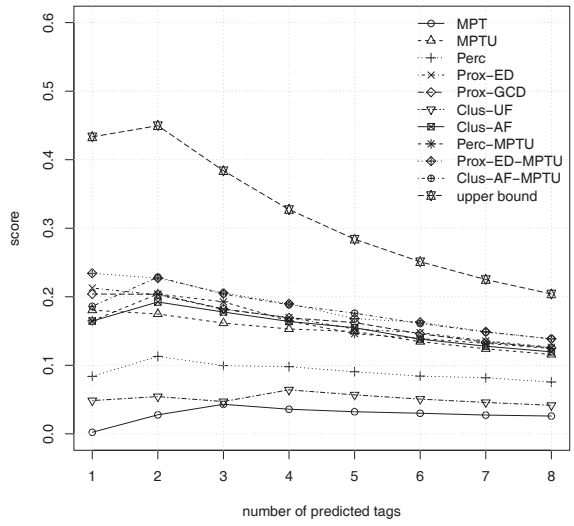


Figure 6: F_1 measure for WideNoise Plus.

are obtained for the recommendation of two and three tags respectively, since the average amount of assigned tags in the dataset is 2.45. Compared to the baselines, we observe, that all algorithms successfully outperform the most popular tags recommender, but also – comparing to the theoretical upper bound – that there is plenty of room for improvements.

An interesting results is that the personalized MPTU approach yields a very good score. It is already better than the computationally intensive Perc approache, but slightly worse than Clus and Prox. It is very interesting that in comparison to the use of the Euclidean distance, the great circle yields almost the same results. For our scenario, this is good news, as similar recommendations are produced with less computational effort. The use of clustered locations (i.e., Clus) yields similar results as Prox-ED and Prox-GCD, but is computationally less expansive.

Looking at the hybridization results, we see that all algorithms profit from the merge with MPTU. Prox-ED benefits far more from MPTU than Clus-AF and achieves the best results among all investigated algorithms – approximately already half of the maximal possible score.

To evaluate the suitability for mobile devices we measured the runtime it took each recommender to predict the tags for the whole evaluation dataset. Figure 7 depicts the computation time for every algorithms[5]. The computational effort of the great circle distance is not acceptable considering the almost same performance. While Prox-ED-MPTU achieved the best recommendation quality, it requires a lot of computation time. Still one has to consider that the analysis was conducted on a relatively powerful computer and that the times will increase with a growing dataset. The runtimes can therefore only be used as indicators, since smartphones have much less computation power and would therefore take much longer.

[5]The evaluation was conducted on a Lenovo ThinkPad X220 with an Intel Core i7-2640M (2.80 GHz), 8 GB RAM, Windows 8 Professional 64-bit, and Gnu R 2.15.3 64-bit.

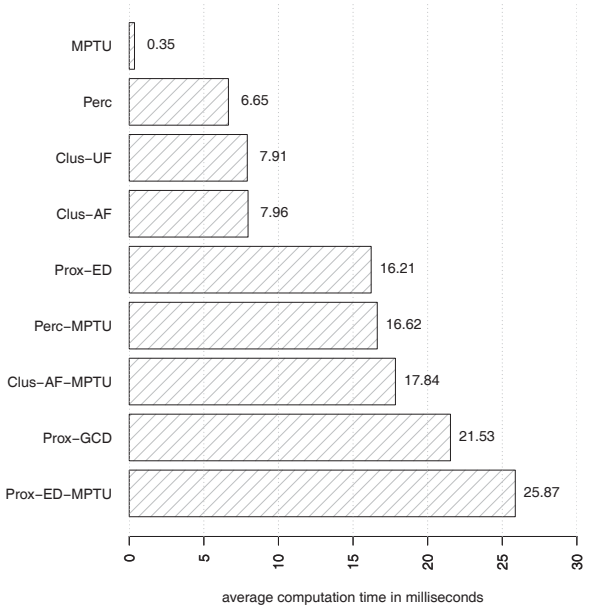


Figure 7: Average recommender runtime.

6. CONCLUSIONS

With the SensorFolkSonomy scenario we have introduced a new task for recommender systems. It requires new models and adaptations of known tag recommendation approaches. WideNoise Plus represents such a SensorFolkSonomy application, and we could compare several algorithms exploiting different aspects of the app's records.

Our evaluations show that the best results are achieved by combining the recommendations of the most popular tag by user recommender and a location proximity-based approach. In future work we plan to evaluate the real runtimes of the algorithms on mobile devices and to adapt other recommendation algorithms to our scenario.

Besides that, there are some additional questions that we want to address concerning the cluster-based approach. Currently, we use k-Means to compute the cluster. It will be of interest to compare varying numbers of cluster or other cluster algorithms and evaluation their performance. Additionally, we will evaluate the performance of further hybrids using machine learning to find weighted combinations with more than two approaches involved.

7. ACKNOWLEDGEMENTS

Part of this research was funded by the European Union in the 7th Framework programme EveryAware project (FET-Open).

8. REFERENCES

[1] R. Abbasi, M. Grzegorzek, and S. Staab. Large scale tag recommendation using different image representations. In *Semantic Multimedia: 4th International Conference on Semantic and Digital Media Technologies, SAMT 2009*, volume 5887 of *Lecture Notes in Computer Science*, pages 65–76. Springer, 2009.

[2] M. Becker, S. Caminiti, D. Fiorella, L. Francis, P. Gravino, M. Haklay, A. Hotho, V. Loreto, J. Mueller, F. Ricchiuti, V. D. P. Servedio, A. Sîrbu, and F. Tria. Awareness and learning in participatory noise sensing. *PLOS ONE*. under review.

[3] M. Becker, J. Mueller, A. Hotho, and G. Stumme. A generic platform for ubiquitous and subjective data. In *1st International Workshop on Pervasive Urban Crowdsensing Architecture and Applications, PUCAA 2013*, pages 1175–1182. ACM, 2013.

[4] P. Bolettieri, A. Esuli, F. Falchi, C. Lucchese, R. Perego, T. Piccioli, and F. Rabitti. CoPhIR: a test collection for content-based image retrieval. *CoRR*, abs/0905.4627v2, 2009.

[5] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, Nov. 2002.

[6] D. Cuff, M. Hansen, and J. Kang. Urban sensing: Out of the woods. *Communications of the ACM*, 51(3):24–33, Mar. 2008.

[7] J. Gemmell, T. Schimoler, B. Mobasher, and R. Burke. Hybrid tag recommendation for social annotation systems. In *19tg International Conference on Information and Knowledge Management, CIKM 2010*, pages 829–838. ACM, 2010.

[8] M. Gupta, R. Li, Z. Yin, and J. Han. Survey on social tagging techniques. *ACM SIGKDD Explorations Newsletter*, 12(1):58–72, Nov. 2010.

[9] M. Haklay. Citizen science and volunteered geographic information: Overview and typology of participation. In *Crowdsourcing Geographic Knowledge*, pages 105–122. Springer, 1. edition, 2013.

[10] A. Hotho, R. Ulslev Pedersen, and M. Wurst. Ubiquitous data. In *Ubiquitous Knowledge Discovery*, number 6202 in Lecture Notes in Computer Science, pages 61–74. Springer, 2010.

[11] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 4702 of *Lecture Notes in Computer Science*, pages 506–514. Springer, 2007.

[12] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in social bookmarking systems. *AI Communications*, 21(4):231–247, Nov. 2008.

[13] N. D. Lane, E. Miluzzo, Hong Lu, D. Peebles, T. Choudhury, and A. T. Campbell. Survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140–150, Sept. 2010.

[14] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Probability and Mathematical Statistics. Academic Press, 1. edition, 1979.

[15] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *3rd ACM International Conference on Web Search and Data Mining, WSDM 2010*, pages 81–90. ACM, 2010.

[16] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 1. edition, 2011.