Cooperative Techniques Supporting Sensor-based People-centric Inferencing

Nicholas D. Lane[1], Hong Lu[1], Shane B. Eisenman[2], and Andrew T. Campbell[1]

[1] Dartmouth College, Hanover NH 03755, USA
{niclane,hong,campbell}@cs.dartmouth.edu
[2] Columbia University, New York NY 10027, USA
shane@ee.columbia.edu

Abstract. People-centric sensor-based applications targeting mobile device users offer enormous potential. However, learning inference models in this setting is hampered by the lack of labeled training data and appropriate feature inputs. Data features that lead to better classification models are not available at all devices due to device heterogeneity. Even for devices that provide superior data features, models require sufficient training data, perhaps manually labeled by users, before they work well. We propose opportunistic feature vector merging, and the social-network-driven sharing of training data and models between users. Model and training data sharing within social circles combine to reduce the user effort and time involved in collecting training data to attain the maximum classification accuracy possible for a given model, while feature vector merging can enable a higher maximum classification accuracy by enabling better performing models even for more resource-constrained devices. We evaluate our proposed techniques with a significant places classifier that infers and tags locations of importance to a user based on data gathered from cell phones.

1 Introduction

Commercial off-the-shelf mobile devices with embedded sensors (e.g., iPhone, Nokia5500, Motorola PSI) are increasingly common in today's market, and have become a focus of people-centric application development due to their growing ubiquity. In this role, devices are owned by individuals rather than residing in a common administrative domain, and data sourced by sensors on the devices may only be available locally (i.e., no centralized repository with common access). Often, the individually tailored models supporting these new applications are based not only on the raw sensor inputs (e.g., camera, microphone, accelerometer), but also on higher level inferences (e.g., location, activity, mood) drawn from particular sensed data features. Identifying the appropriate data features and best performing models continues to be a subject of intense interest in support of these new applications [18] [16] [17] [15].

Against this backdrop, two main challenges facing the construction of accurate inference models are the lack of appropriate data inputs and the time

and effort that must be spent in training a model of sufficient accuracy. The consumer-device-based sensing substrate upon which people-centric applications are built is characterized by heterogeneity in terms of sensing and other resources (e.g., memory, battery capacity). Therefore, the data inputs most useful in generating high accuracy models are not likely to be available on all devices. As an example using a snapshot of current technology, classifiers distinguishing indoor vs. outdoor locations are built using data features from GPS and WiFi sensors [16]. However, GPS and Wifi are integrated into only a relatively small percentage of cell phones on the market today. This heterogeneity often requires users of less capable devices to settle for less accurate models based on other available data features. Figure 1(a) illustrates the result of this situation, showing the experimental performance of a significant places classifier (see Section 4 for implementation and performance details) for four device capability classes (CC): CC1 is Bluetooth only, CC2 is Bluetooth and WiFi, CC3 is Bluetooth and GPS, and CC4 is Bluetooth, WiFi and GPS. Perhaps unsurprisingly, the accuracy of location recognition increases as the sensor inputs from more capable cell phones are used to generate better models. These observations motivate and inspire our proposed *opportunistic feature vector merging* approach with which we seek to push the model performance possible with lower tier devices (e.g., CC1) towards that possible with higher tier devices (e.g., CC4). With feature vector merging, data features from more capable devices are borrowed and merged with data features natively available from a less capable device in the model building stage, allowing the less capable device to generate a higher accuracy model. This borrowing is facilitated by opportunistic interaction (though not necessarily communication), both direct and indirect, between a less capable device and a more capable device in situ. As an example of direct interaction, as two cell phone users follow their daily routines, the cell phone without GPS can borrow GPS data features from the cell phone with GPS as an input to its indoor/outdoor model. An indoor/outdoor model based on GPS feature instances borrowed over a period of time may also be built. In the indirect interaction case, both devices collect data samples according to their respective capabilities. Subsequently, centralized matching between commonly collected features (i.e., not GPS) may provide for a binding between the feature vector collected by the phone without GPS and the GPS features collected by the GPS-equipped phone. The GPS features can then essentially be borrowed via this binding.

Even when devices provide an appropriate set of data features to build accurate models, users may be required to gather a large set of training data (perhaps manually labeling it) before applications using the model outputs work at their peak level. The inconvenience in both the labeling of training data and the time required for model training to complete may act as disincentives to the broad-scale adoption of new people-centric applications [16]. One approach to reduce model training time and effort is to support the sharing of labeled training data among users. Sharing training data has the effect of reducing the per-user training time and labeling effort when building the necessary collection of training data, but is also likely to reduce the accuracy of the resulting models. This is es-

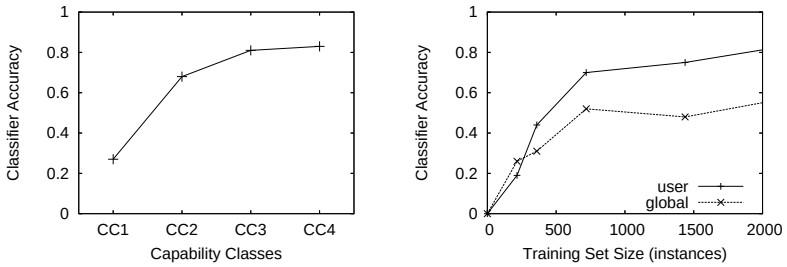


Fig. 1: Classifier performance relative to varying device capabilities and the size of the training data used. In (a), accuracy is plotted for various capability classes (CC): CC1 is Bluetooth only, CC2 is Bluetooth and WiFi, CC3 is Bluetooth and GPS, and CC4 is Bluetooth, WiFi and GPS. In (b), accuracy is plotted against the training set size.

pecially true in people-centric sensing systems based on common mobile devices like cell phones. In this context, sensor data features are often limited by the non-ideal set of sensors embedded or interfaced to the cell phones, and also the quality of the training process is difficult to control. Therefore, models in this domain are often more tightly bound to the individual in order to achieve higher accuracy. Consider Figure 1(b), which shows the classification accuracy versus training set size for our significant places classifier. The dashed line curve reflects the accuracy of a model built by merging experimental training data from all participants (see Section 4 for details). The solid line curve in Figure 1(b) shows the average accuracy of a collection of models, built on a per-user basis using only data sourced from each respective participant. For a given value A on the x-axis, for the per-user models, each of the N users provides A instances, while for the global model each user contributes roughly A/N instances. The quantity of per-user training data required in building the global model is low since model training cost is amortized over all the users in the system. However, the accuracy is also consistently low due to the aforementioned problems with global training data sharing in people-centric sensing systems. With our proposed *social-network-driven sharing*, we provide a hybrid approach that builds models based on training data shared within social circles, within which we conjecture group vocabularies and other commonalities lead to more consistently labeled training data and a higher model accuracy, while still reducing the quantity of per user training data required.

The contributions of the paper are: (i) we are the first to propose opportunistic merging of feature vectors between devices to improve model accuracy on lower capability devices; (ii) we propose the sharing of training data and models between devices by leveraging the social relationships between their users; and (iii) we implement and test these two complementary techniques in the context of "significant places" [21] [2] [9], a people-centric service targeting sensor-equipped mobile devices.

2 Related Work

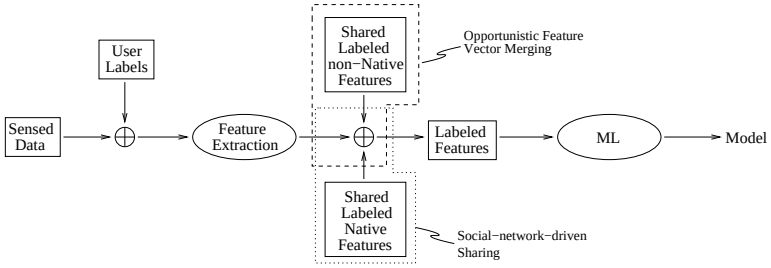
The problem of acquiring suitably labeled training data to build classification models is well recognized, and is addressed in the literature in a number of ways. To the best of our knowledge, there is no existing research targeting feature sharing through opportunistic interaction. This may be due to the fact that feature sharing may add uncertainty to the system and is thus a counter-intuitive approach to improving model accuracy. Opportunistic sharing of data features and models can be viewed as a special case of opportunistic data exchange more generally. As such, sensor fusion [14] in ephemeral proximity-based networks is related, though neither communication within socially connected groups nor the constraints and advantages of sharing to enhance classification accuracy are treated in the general case. Sharing training sets from one user's model to improve the performance of another can be thought of as co-training [22].

The Tapestry system [7] uses a collaborative approach to perform document filtering (e.g., email) based on the reactions/responses of others. The authors of [10] propose what they term collaborative machine learning, which unifies collaborative filtering and content-based filtering. The approach considers both the user's data content, as well as attributes and descriptors, to gain a better idea of the similarities among users, providing better accuracy for document retrieval and recommendation applications. A similar sharing concept is explored in [19], where the authors propose a method for recommendation sharing based on statistical correlations in users' data sets (e.g., music artist playlist). While these approaches enable sharing of what can be considered model training data or classification models, the sharing ignores social group connections. We conjecture our social-network-driven sharing proposal can be integrated into these systems to improve performance (e.g., in Tapestry, only considering annotations created by members of the same social group). Using social connections to guide sharing can be thought of as semi-supervised learning [22].

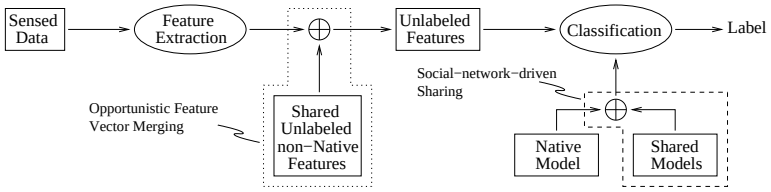
There are a number of research papers contributing to various aspects of "significant places" applications, including significance learning [13], location clustering [21], and location prediction [2]. We use significant places, representative of emerging applications using sensor-enhanced inferences and targeting mobile devices, to demonstrate the usefulness of our techniques of opportunistic feature vector merging and social-network-driven training data sharing.

3 Proposed Techniques

At a high level, a standard approach to building models involves first sensing available data, extracting and labeling sensed data features that accurately describe states, and then finding a classification technique that provides high accuracy and high confidence classification. In terms of model usage, first the available data is sensed, the necessary features are extracted and fed into the model without labeling, and then the model outputs the inferred label. In Figure 3, we represent these two processes pictorially, and include the stages in each



(a) Model learning process.



(b) Model usage process.

Fig. 2: Typical model learning and usage processes, and how opportunistic feature vector merging and social-network-driving sharing hook into these. In the diagrams, the circumscribed "plus" symbols represent a merging of information (e.g., labeled features). Actions are enclosed ellipses, while objects are enclosed in rectangles.

process where our proposed techniques hook in (encircled with dashed/dotted lines). As indicated in the diagram, feature merging and social-based training data and model sharing are complementary techniques that can be composed in both the model learning and usage processes to improve performance. In the following, we discuss in more detail a number of design and implementation challenges, providing a roadmap for future work needed to realize the full potential of our approches. We begin to address these challenges in this work.

3.1 Opportunistic Feature Vector Merging

With opportunistic feature vector merging, we aim to leverage opportunistic interactions (both direct and indirect) between devices with different capabilities to improve the model accuracy achievable on less capable devices. Less capable devices borrow from more capable devices features that allow for the generation and subsequent use of more accurate models than those possible to generate from only natively available data features. Here, the capability of the device can be thought of in terms of sensor configuration, available memory, and CPU/DSP characteristics. Thus, as in the example given in Section 1, opportunistic feature merging can provide desirable vector elements (e.g., those derived from GPS and WiFi data) that are not available natively due to the sensor configuration. Secondly, merging can provide additional data features of native types that may be needed, for example, when the device is not capable of storing a time series

of the required size. Finally, opportunistic feature merging can be used to share features extracted from external data that are also available natively, but can not be calculated on the device due to device limitations (e.g., a computationally intensive FFT of microphone data cannot run on a CPU-limited device even though the device has the microphone). A number of questions arise when considering a system design that uses opportunistic feature vector merging, which we explore in the following subsections.

Determining what features are sharable. Given the mobile devices available on the market today, the following hardware sensors are available in at least a subset of devices: camera, microphone, accelerometer, 802.11 radio, Bluetooth radio, GPS receiver, cellular radio. The raw sensor data from each of these sources can be processed in many ways, alone and in combination, to extract features useful for model building. However, not all of the features are equally sharable for opportunistic feature merging. For example, two co-located devices, one with a GPS receiver and with an 802.11 radio can likely exchange features from these sensors for mutual benefit. On the other hand, data mined from a user's calendar on one device may not be of much use on another user's device, and may even result in a worse model for the borrowing device. Similarly, raw samples from light sensors separated by even a very small distance by have very different values due shadow patterns, and may not be amenable to sharing. However, it may be useful to share temperature samples even at longer distances since temperature gradients tend to be shallower. While determining which particular features are beneficial to share or not likely depends on the classifier (e.g., how susceptible is the output to inaccuracy in the input), a reasonable guideline is to only share features that are not highly person, device, or location specific. Even if these contribute to a better classifier on their native device, they are unlikely to do so on another user's device.

What are the feature sharing mechanisms. In Section 1, we introduce two types of opportunistic feature merging, depending on whether the interaction is direct or indirect. The feature sharing mechanism for each variant is slightly different. For direct interaction, devices periodically broadcast their available data sources (e.g., hardware sensors) via an available short range radio interface. Advertising only the data sources is preferable to advertising the entire feature set in terms of efficiency, since there are likely many possible features per data source. Additionally, only those data sources that are likely to be sharable (as discussed previously) should be advertised to reduce unproductive feature sharing. Devices that are interested in borrowing reply with a request for all the features available for a given (set of) data source(s). Requesting only the features, rather than all the raw data, saves on communication energy spent by both the lender and the borrower. With direct interaction, models can be used in a distributed way on each mobile device. Over time, it is also possible for a device to collect enough shared feature instances to build models based on shared features, potentially allowing for infrastructureless bootstrapping of the system.

In contrast, feature merging via indirect interaction uses a centralized approach, requiring no direct device peer interaction. All devices collect samples, extract features, and generate models to the best of their respective abilities in situ. Subsequently, when each device transfers its training data/features to a dedicated server, the merging process looks for evidence in the features provided by all users that two or more devices were sensing the same location or event. If so, then these devices are able to share features to generate better models. Indirect sharing is helpful if two devices are co-located but can not communicate locally due, for example, to radio incompatibility. Indirect sharing also allows devices that sense the same event/phenomenon but are never co-located to share data, if the sensed event/phenomenon is relatively constant in the time between the respective devices' visits. Finally, indirect sharing potentially saves on communications costs over direct sharing since no local data exchange is necessary. For example, consider two devices that each have a GPS receiver, but only one has a CO_2 sensor. In this case, the merging process can identify through matching GPS readings that the devices were in roughly the same place at the same time. Then the device without the CO_2 sensor can borrow the CO_2 readings and incorporate them into its training data to generate improved models.

What to do when shared features are not available. One drawback to building models requiring borrowed features is that there is no guarantee a device will be on hand to share the required features when the model is to be used. We address this with two approaches. First, each device generates a collection of models, each relying on different sets of available features. The device uses the model that has the best expected performance (i.e., w.r.t its confusion matrix) given the features available at the time of classification. In the worst case, this will be the model learned only from device-native sources. Second, we build models using algorithms that are more resilient to missing or noisy elements of the feature vector. For example, the KNN imputation method performs better relative to the comparison technique of the LNN classifier [1].

Privacy concerns in sharing. Opportunistic feature sharing potentially leaks personally sensitive information (e.g., location trace). One option is to provide the user with the ability to configure the type of data that is sharable, and with whom. Another option is to share features without including any identifiers in the packet payload. However, for direct sharing the MAC address of the short range radio used to share can be logged. Use of disposable MAC addresses is possible [8], but this may limit functionality for certain PHY/MAC technologies. Providing truly anonymous data exchange for ad hoc mobile devices is a focus of ongoing research in the community [5], but is outside the scope of this paper.

3.2 Social-network-driven model and training data sharing

With social-network-driven training data sharing and model sharing, we aim to leverage social connections between device users to reduce the amount of time

and effort an average user must expend to train her models while maintaining reasonable model accuracy. These social connections may be short-lived or persistent, and include connections based on proximity, professional groupings, family, friends, people sharing common interests (e.g., tango class), and many others. A number of techniques for mining social graphs from various information sources exist, but a review of this literature is out of scope. In the following, we discuss training data sharing, deferring treatment of model sharing to a later section. As discussed in Section 1, sharing training data generally has the effect of reducing the time and effort of training, but has the undesirable side effect of reducing the accuracy of models generated with this mixed data. Features that have good discriminative power within particular population subgroups, lose effectiveness within larger groups. We propose to allow sharing only within social circles to moderate this reduction in accuracy, while still reducing training time and effort. In the following, we describe a number of challenges to social-network-driven sharing, and discuss the motivation of model sharing between members of the same social group.

Exploiting social connections. Previous work [4] [6] suggests ways to mine sensor-based and other data to infer social graphs where the vertices are people or groups and the edges are relationships. Assuming known social graphs, we construct models with training data sourced on the basis of the strength of social connections (edges in the social graph) between the intended target of the model (e.g., the device user) and others. A lower bound on the strength of connection between two users may apply such that sharing does not occur below this threshold. We expect people who are members of the same social groups (such as combinations of cultural, workplace, social, or family groups) will have similar background or other context that translates into similarity in label definitions (i.e., what classes are important and what are the appropriate labels). By exploiting awareness of the social connections between people we build a training set sourced by a variety of people that still produces a model for a particular individual (or group) that approximates the performance of a model built solely from training data sourced from this individual (or group) in terms of both classification accuracy and the understandability of labels.

A number of interwoven social graphs are likely to apply to a given set of individuals. The nature of the inference problem (i.e., the application, or learning technique) may determine which social graph to use when considering which training data to import from other users. In the context of our running example of significant place classification, if a user may provide free-form labels (e.g., colloquial labels for locations), it may be appropriate to incorporate labeled instances from other nodes in her social network with whom she is frequently physically located, under the supposition that a location-specific vocabulary is likely in use (e.g., workplace vernacular, regional dialect). On the other hand, individuals to whom one is extremely close socially (e.g., a girlfriend), may be of less use in sharing location-specific vocabulary if they are frequently physically

distant. Similarly, labeling of certain activities or social settings may be more culturally and demographically driven.

Quality and consistency issues. A number of challenges arise related to the quality and consistency of shared data instances.

First, the quality of the training instances may vary from user to user due to the care taken when the training data was gathered, the training methods used, and the training environment (e.g., data collected under non-typical circumstances can lead to a model that does not perform well in general). Challenges in repairing ill-labeled data aside, it is difficult even to determine which instances are lower quality. This is especially difficult when the pool of available labels is small and statistical techniques such as anomaly detection are not applicable. Because of this, importing lower quality training data can pollute one's natively collected data, leading to poorer model performance.

When free-form labeling is used, opinions may vary among users on the proper size of label set, the feature support of each label, and the label itself. A related complication is that lexicographically identical labels may mean different things to different people and different labels may mean the same thing to different people. One way to address these issues is to apply structure to the labeling stage such that a fixed set of valid labels, each with a provided definition, is imposed on all users. However, this approach restricts the classifications problems that can be solved, and may result in a model that, though accurate, gives labels that are not well understood by a given user.

Designing models robust to mixed source data. Given the lack of flexibility of structured labeling, we support free-form labeling. While sharing within social circles mitigates labeling consistency issues to some extent, the process of learning models must still be robust to them. Incorporating contradictory instances, where the same class of features is given two or more different labels (by multiple users), leads to a situation where the same class of feature vectors will be mistakenly fragmented into multiple labels. (The impact of this fragmentation is somewhat problem-specific, since a classifier that seeks only to differentiate between logical classes of might perform well even with fragmented features.) We use an unsupervised clustering approach to detecting and correcting this fragmentation in our significant places implementation discussed in Section 4. Instances can then be appropriately grouped regardless of their label, with the introduction of some error due to imperfect grouping. After clustering, a normative label may be applied for consistency.

Social-group-based Model sharing. In addition to sharing training data, the models themselves are also candidates for sharing. The trigger for borrowing models would be noticing that the performance (e.g., recall, precision) was better in the model of a fellow social group member than in yours for the same feature vector. In this case, either the user's device can check neighboring devices in

situ to if they have an appropriate model with better performance (e.g., via an advertise-request-response protocol), or the model sharing can be done in a centralized way on a dedicated server. The rationale for model borrowing between members of a social group in particular is that even though the models may have been learned based on training data labeled by your buddy, your buddy's labels are likely to make sense to you because of your shared membership in the social group. Elements of shared models that may be particularly helpful in improving a user's locally generated model can be permanently incorporated by importing the appropriate training data and relearning the local model. This is beneficial in the online case since performance can be maintained even when the neighbor with the better model is not nearby, and in the offline case it reduces unnecessary processing.

4 Evaluation

To evaluate the impact of opportunistic feature vector merging and social-network-driven data and model sharing on a real people-centric application, we implement a version of the "significant places" classifier (e.g., [11] [6]). We use this as a vehicle to demonstrate the application of our techniques. In the following, we describe the implementation and focus of our variant of significant places, and the experimental data collection methodology, followed by selected performance results.

4.1 Significant Places

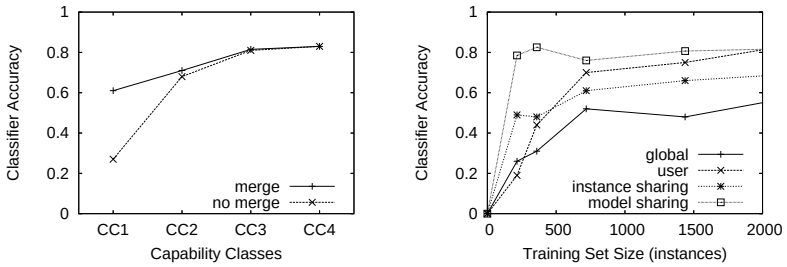
A frequently examined classification problem in the literature is that of taking location traces of a user and distilling them into a sequence of visits to places that are significant to her (e.g., home, work, gym). This is used by applications that present historical summaries of the user's daily life [2], or even to determine when a person has taken a wrong turn heading toward home [18]. A generic significant places classifier may be thought of in terms of three main phases. In the first phase, various data features (e.g., visitation frequency and dwell time) of a user location trace are extracted from the raw data and analyzed to identify locations and infer whether they are significant to the user. In the second stage, the significant places are labeled, either by mapping the location feature vector to a set of system-provided labels or by manual prompting of the user to allow for personalized labels. In the third phase, the classifier is run to see how accurately the system can recognize that a user has entered a significant place. A number of proposals (e.g., [21] [11]) exist addressing the first phase of learning models to infer significance. As significance inferencing is orthogonal to our techniques, in our implementation we simplify the first two phases and have the user manually label instances of location feature vectors as significant or not (c.f. the collection methodology in Section 4.2). Based on these labeled instances, we then evaluate the impact of our merging and sharing techniques on the accuracy and label understandability of models built to recognize the labeled significant places.

4.2 Data Collection Methodology

As the sensing, processing and display capabilities of cell phones increase, cell phones provide a unique chance for researchers to understand the real mobile user behaviour and to provide true in situ mobile services. To gather user-labeled significant place instances we use Nokia N80 and N95 smart phones. Both models feature Bluetooth and an 802.11g WiFi interface. The N95 also comes with an integrated GPS receiver. To facilitate user labeling of significant places, we implement and install a PyS60 (Python for Symbian S60) client on each cell phone. The client provides two fundamental services: user labeling and daily trace recording, and sensor sampling. For each significant place, the user enters a new label (or selects a previously entered one). With a button click, the user indicates when she enters and leaves the selected significant place. The client records the label, and the enter and leave times for each significant location visit. From these entries, the client generates a significant location trace for each user. The user is able to review and edit the daily trace to verify its correctness. The sensing daemon runs in the background to sample from the Bluetooth, WiFi and GPS, if available. We use an inter-sampling interval of approximately three minutes, which gives an average battery life of more than 6 hours. The sampling duration is lasts between 30 and 60 seconds, depending on how long the function call to scan the Bluetooth neighborhood takes to return. The following data are captured: GPS - latitude, longitude, altitude, accuracy, time, speed, number of satellites; WiFi - beacon interval, security mode, SSID, BSSID, signal strength; Bluetooth - address, device name, service type.

4.3 Data Analysis Methodology

The inputs to the models we construct are based on a feature vector formed from three types of elements; location, time and social context. Clock, GPS, WiFi and Bluetooth data give rise to the following features, which are also further processed to generate averages and variances. From the clock, we extract day/night, 3-hour block, the duration of visitation, weekday/weekend and, business/after hours. From WiFi, we extract the absence or presence of access points (APs) identified by their MAC addresses, the relative RSSI order among the visible APs, the individual and aggregate RSSI, and other AP statistics that have been previously used to distinguish geographic locations [9] [12]. From social context, we seek to capture the social characteristics of the location. We extract the number of Bluetooth-toting people in the area (assuming one device per person). This is used in concert with a list of the people with whom the individual has social connections (e.g., from Facebook or other social networking sites). Use of Bluetooth and WiFi features allows us to distinguish between adjacent locations that may have very similar GPS features. We use the Weka machine learning workbench [20] for our analysis, specifically the default configuration of the bagging algorithm applied to the decision tree module, REPTree. All models are trained on a randomly selected 50% of the data set available for it. In the following, we describe initial performance results achieved with models based on



(a) Feature merging. (b) Social sharing.

Fig. 3: Performance Plots

a proof-of-concept implementation of our sharing and merging ideas. We leave a deeper investigation of the design space for later work.

4.4 Performance Results

In an experiment run over 12 days, data we collect from 13 phone users (four Nokia N95 and nine N80 cell phones) using the collection methodology outlined above comprises 14375 labeled instances of 62 uniquely labeled locations. We run post-collection validation via manual checking and participant interviews to verify the integrity of the data set. All phone users are members of, or are socially connected to the Computer Science Department at Dartmouth College. Participant ages range from 24 to 49; one user of the 13 is female. We also gather results from a survey (described subsequently) that includes the 13 phone users and an additional 8 survey-only participants. These latter fall in the same aforementioned age range and have the same departmental connection; one of the additional 8 is female. We present results demonstrating the potential impact of the opportunistic feature vector merging and social-network-driven sharing.

Feature Vector Merging Performance. We generate models, "merge" and "no merge", from experimental data, and examine the impact of performing direct sharing of features based on different device capabilities. Sharing of features is done on the basis of Bluetooth connectivity. Whenever two devices in the experiment detect each other in their Bluetooth neighborhood then feature sharing is enabled. An exchange of feature vector elements occurs when possible, giving participant nodes a richer feature vector they would otherwise have based on native sensors. Although all Nokia N80 and N95 phones have WiFi and all N95 phones have GPS, to emulate four distinct capability classes of devices for some devices in the experiment the WiFi on the N80 phones and the GPS on the N95 phones is ignored as needed to support allowing four different classes to be emulated. We build the models as follows: a single model is generated for each user during the evaluation. This model is trained using all the feature vectors available, even those that are intermittently available via sharing. This

results in numerous feature vector instances with missing elements, since sharing is not continuously available. We do not explicitly handle missing data within execution of our model (e.g., using a model swapping technique), but instead use a machine learning technique (bagging) innately robust to the missing data. Models are built on a per-user basis using training data specific to the user and user's device, and based on his or her own opportunities for merging.

Figure 3(a) shows a comparison of these two models. It reports the average classification accuracy for each of the per-user models generated. Accuracy is plotted against the phone capability class. In each of these classes the performance is reported for all phones being limited to this operating level or lower. The plot shows that with feature sharing, we can always gain an advantage in model accuracy, except for capability class CC4 devices since those already have all the sensors natively. In our campus environment and predominantly indoor significant places, WiFi is the most powerful feature to share to improve accuracy, as indicated in the large increase between CC1 (Bluetooth only) and CC2 (Bluetooth and WiFi). In environments where WiFi features are less available, we expect shared GPS-based feature elements to be the most helpful.

Social-network-driven Sharing Performance. We generate four models from experimental data, including two that incorporate sharing to support model generation, "instance sharing" and "model sharing", and two that do not, "global" and "user". With these models, we investigate the impact of sharing on classification accuracy with respect to the amount of training data provided by each user. In all cases, models are trained using a randomly selected 50% of the data, with the balance used for performance testing. The device population comprises the following mix of capability classes: 5 Bluetooth only; 4 Bluetooth and WiFi; and 4 Bluetooth, WiFi and GPS.

As discussed in the Introduction, the "global" model is generated by pooling training data from all participants, with each user contributing roughly the same amount to the pool. The "user" model is generated on a per-user basis using only training data sourced from the user herself.

The "model sharing" approach generates per-user models as in the "user" approach, but then multiple models are tested before settling on a label output for a particular instance. The decision to apply another model or settle on the current result is based on estimated accuracy for the generated label. The choice of whose per-user models to choose for a given classification task is driven by social connections between users, prioritizing social connections that are logically related to the classification task. In so doing, models are applied according to a hierarchy of social groupings. Users' models within a group are ranked arbitrarily in our implementation, but the strength of personal social ties within a group can also be considered when deciding the order of model application. The application of models terminates either when a confidence threshold is reached (to improve classification accuracy), or if a certain maximum number of models is applied (to limit overhead). Lastly, with "model sharing", we always test the "global"

model (global sharing) as well, and the result with the highest confidence among all the tested models becomes the final output label.

With "instance sharing", per-user models are built, but for a given user the training data is sourced from the user and from people within the user's social networks. As with "model sharing", a social group hierarchically is constructed considering the purpose of the classifier and the groups' potential impact in this regard, and intra-group ranking is also handled in the same way. In "instance sharing", training data instances are accumulated iteratively, considering one user per step, until the overall required number of training instances L is assembled. At each step i, the user is tapped to provide up to L/i instances. The goal, if K steps are taken to accumulate L instances, is to have each user provide L/K instances. At any point, if a user can not contribute the desired L/i instances, randomly chosen instances from the global pool are chosen, but are removed from the overall required L if they are no longer needed as filler.

The social groups present in our experimental user population and used for the sharing-based models are: "students", enrolled students at any college; "Dartmouth", enrolled students at Dartmouth College; "batch", grouped according to the year arriving at Dartmouth; "founders", founding members of SensorLab that have worked together since the inception of our research group; "SensorLab", all members of the SensorLab research group; "CMC", all members of the CMC Lab research group; "Chinese", have a strong social tie including a daily lunch group; "Facebook", social connections as defined within Facebook; "basketball", participants in a local summer basketball club; "town", those with a common town of residence; "European", those with a European origin; "non-U.S.", those with any non-U.S. origin. We order these groups according to the degree to which we expect them to improve our significant places classifier. Participants in the study are members of multiple different social groups.

The rationales for a few of the ranking decisions are as follows. We rank "Facebook" above "Dartmouth" since we expect Facebook friends to use common names for specific locations more so than does the general pool of Dartmouth students. The same logic leads us to rank the "members" ahead of "students". Group rankings may fluctuate seasonally. For example, the "basketball" group meets regularly, but only during the summer - familiarity (e.g., common experience, shared stories, shared vocabulary) decays over the rest of the year. Conversely, the "Chinese" group meets every day at noon for lunch, so the social ties remain strong throughout the year.

Figure 3(b) shows the classification accuracy versus training set size for each type of model. For a given value A on the x-axis, for the per-user models, each of the N users provided A instances, while for the global model each user contributes roughly A/N instances. For the sharing-based models, we assign equal weights to all social groups; the group hierarchy is flat. Each of the M users involved (i.e., through common social group membership) contributes A/M instances. The figure demonstrates the advantage of sharing only within social groups rather than globally as the model accuracy curves for both instance sharing and model sharing are always above that of the global model. Addition-

ally, we see the advantage in terms of model learning time (i.e., required training data set size) that both instance and model sharing provide. We find that social-based model sharing achieves a higher maximum accuracy than training instance sharing for our data set. As expected, the per-user model outperforms instance sharing as the amount of available training data becomes large enough.

Classifier Performance Details Figure 3 shows the sensitivity of the model accuracy to both the richness of the feature vector and the availability of training data, accuracy alone does not provide the full picture of the model performance. In Table 1, we present additional performance details (true positives rate (TPR), false positives rate (FPR), precision and recall) for each of the classifiers we use (i.e., the average performance across all classes). For the results shown here, the training set size is fixed at 719 instances.

Model	User (avg)	Global	Feature Merging	Model Sharing	Instance Sharing
TPR	0.598	0.383	0.617	0.721	0.389
FPR	0.563	0.349	0.565	0.652	0.304
Precision	0.691	0.496	0.712	0.807	0.544
Recall	0.598	0.383	0.617	0.721	0.389

Table 1: Classifier statistics.

4.5 Survey Results

To understand the impact of social-network-driven model and instance sharing on the understandability and appropriateness of the model output labels, we survey 20 participants concerning the outputs of the sharing-based models used in the experiments described in Section 4.4. In this survey, we focus on determining the participants' depth of understanding of shared labels, and their feeling of the appropriateness of these labels when shared socially.

In Table 2, we report statistics on the level of comprehension people from different social groups have of labels produced by members of their own versus other social groups. Survey participants are asked questions to determine their level of understanding regarding 8 different labels. For each label, users are asked to identify to where they think a label refers when given the label provider's name and the label itself. The understanding is categorized as "strong", "weak", or "none" depending on how accurately the label is positioned on a map; "strong" if the exact location is indicated, "weak" if a location in the vicinity is indicated, and "none" otherwise. Label providers are not asked about their own labels.

Comprehension levels are shown in Table 2 for the dominant social groups ("SensorLab" and "CMC") that produced the most labels in our experiments. Members of the same social group share a better comprehension of each other's labels on average, compared both with members of the other group and the average population. For example, on average members of "SensorLab" stated

Group	Strong		Weak		None	
	"SensorLab" Labels	"CMC" Labels	"SensorLab" Labels	"CMC" Labels	"SensorLab" Labels	"CMC" Labels
"SensorLab"	0.75	0.32	0.09	0.15	0.16	0.53
"CMC"	0.40	0.55	0.05	0.03	0.55	0.43
All	0.48	0.34	0.14	0.31	0.38	0.49

Table 2: The level of comprehension people from different social groups have of labels produced by members of their own or other social groups. Members of the same social group share a better comprehension of each other's labels on average.

they had a "strong" comprehension of 75% of labels generated by members of their own group, but no comprehension of 53% of the labels generated by "CMC" members. These results indicate that a model based on global sharing is likely to perform poorly in terms of understandability, in addition to accuracy (Figure 3(b)), underscoring the importance of social-based sharing.

To determine the statistical significance of the results in Table 2, we run a χ^2 test with a threshold of 0.05, and calculate $\chi^2_\alpha = 5.9915$. First, we test the null hypothesis that comprehension of the labels provided by the "SensorLab" group is independent of group membership. The null hypothesis rejected with $Q = 14.401$. In the analogous test for the comprehension of labels provided by "CMC" members, we calculate $Q = 6.3068$, again rejecting the null hypothesis, concluding that the "CMC" members' better understanding (relative to that of "SensorLab" members) of labels provided by fellow members is statistically significant. These results give statistical credence to the notion of social-group-based sharing.

Table 3 presents results from the same survey on the appropriateness of labels provided by selected individuals for particular places. Given a place, survey participants rate four possible labels (each taken from labels generated by the 13 phone experiment participants) to describe the place on a scale from 0 to 4 (0 means "not appropriate"). Table 3 shows selected results for three (place,label) combinations. Generally, the table shows that the perceived appropriateness of a given label can be strongly impacted by social connections, as reflected in the higher values along the diagonal. For example, at least one member of each of the two laboratory groups ("SensorLab" and "CMC") included in the user set use the label 'Lab' to refer to their respective lab. Members of each lab think this label applies most appropriately to their own lab (i.e., average rating of 2.10 and 1.75 for their own versus 1.25 and 1.00 for the other lab). "Chinese" comprises those that often go together for lunch at the Orient restaurant. The table shows that "Chinese" members are more likely than "SensorLab" members (though not more so than "CMC" members) to find the diminutive 'Ori' acceptable. The lack of distinction between between "SensorLab" and "Chinese" for this label may be due to the existing overlap in group membership. These results support the use of socially shared labels in the significant places test application.

	Place:Label		
Groups	SensorLab lab:'Lab' ("SensorLab")	CMC lab:'Lab' ("CMC")	Orient Restaurant:'Ori' ("Chinese")
"SensorLab"	2.10	1.00	0.83
"CMC"	1.25	1.75	1.25
"Chinese"	0.80	1.20	1.20

Table 3: The level of appropriateness of selected labels as viewed by different social groups. Social connections can strongly impact the perceived appropriateness of a label, an important motivation for social-based instance/model sharing.

5 Conclusion

As the sensing and computation capabilities of commercial devices such as cell phones increase, the development of people-centric applications augmented with sensor inputs will also accelerate. To facilitate the wide-scale adoption of these applications, we have proposed two techniques aimed at both increasing the accuracy of feature classification used by these applications, reducing the burden on the user in terms of providing labeled training data. We have demonstrated the efficacy of both opportunistic feature vector merging and social-network-driven sharing in the context of "significant places", a useful classification process for people-centric sensor-enabled applications. Our results underscore the opportunity and importance of leveraging the inevitable device heterogeneity that results from the evolution of technology, and the importance of taking social relationships into consideration when sharing in support of model building.

Acknowledgment

This work is supported in part by Intel Corp., Nokia, NSF NCS-0631289, and the Institute for Security Technology Studies (ISTS) at Dartmouth College. ISTS support is provided by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

References

1. E. Acuna and C. Rodriguez. The treatment of missing values and its effect in the classifier accuracy. In *Classification, Clustering and Data Mining Applications*, pp. 639–648, 2004.
2. D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.
3. C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

4. T. Choudhury and A. Pentland. Sensing and modeling human networks using the sociometer. In *ISWC '03: Proc. of the 7th IEEE Int'l Symp. on Wearable Computers*, pg. 216, Washington, DC, USA, 2003.
5. L. P. Cox, A. Dalton, and V. Marupadi. Smokescreen: flexible privacy controls for presence-sharing. In *MobiSys '07: Proc. of the 5th int'l conf. on Mobile systems, applications and services*, pp. 233–245, New York, NY, USA, 2007. ACM.
6. N. Eagle and A. S. Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.
7. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.
8. M. Gruteser and D. Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. In *WMASH '03: Proc. of the 1st ACM Int'l workshop on Wireless mobile applications and services on WLAN hotspots*, pp. 46–55, New York, NY, USA, 2003.
9. J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes. Learning and recognizing the places we go. In *Ubicomp*, LNCS 3660, pp. 159–176. Springer-Verlag, 2005.
10. T. Hofmann and J. Basilico. Collaborative machine learning. In *From Integrated Publication and Information Systems to Virtual Information and Knowledge Environments*, pp. 173–182, 2005.
11. J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(3):58–68, 2005.
12. J. Krumm and K. Hinckley. The nearme wireless proximity server. In *Ubicomp*, LNCS 3205, pp. 283–300. Springer, 2004.
13. L. Liao, D. Fox, and H. Kautz. Location-based activity recognition. In *Advances in Neural Information Processing Systems 18*, pp. 787–794. MIT Press, 2006.
14. H. Luo, J. Luo, Y. Liu and S. K. Das. Adaptive Data Fusion for Energy Efficient Routing in Wireless Sensor Networks. *IEEE Trans. on Comp.*, 55(10), pp. 1286-1299, 2006.
15. N. Marmasse, C. Schmandt, and D. Spectre. Watchme: Communication and awareness between members of a closely-knit group. In *Ubicomp*, LNCS 3205, pp. 214–231. Springer, 2004.
16. E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell. Cenceme - injecting sensing presence into social networking applications. In *EuroSSC*, LNCS 4793, pp. 1–28. Springer, 2007.
17. D. J. Patterson, L. Liao, D. Fox, and H. A. Kautz. Inferring high-level behavior from low-level sensors. In *Ubicomp*, LNCS 2864, pp. 73–89. Springer, 2003.
18. D. J. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic, K. Olson, S. Wang, D. Fox, and H. A. Kautz. Opportunity knocks: A system to provide cognitive assistance with transportation services. In *Ubicomp*, pp. 433–450. Springer, 2004.
19. U. Shardanand and P. Maes. Social information filtering: algorithms for automating word of mouth. In *CHI '95: Proc. of the SIGCHI conf. on Human factors in computing systems*, pp. 210–217, New York, NY, USA, 1995.
20. I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques with Java implementations.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.
21. C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen. Discovering personally meaningful places: An interactive clustering approach. *ACM Trans. Inf. Syst.*, 25(3):12, 2007.
22. X. Zhu. Semi-Supervised Learning Literature Survey. In *Tech. Report UW-Madison 1530*, 2005.