Towards Population Scale Activity Recognition: A Framework for Handling Data Diversity

Saeed Abdullah
Cornell University
Ithaca, New York, USA

Nicholas D. Lane
Microsoft Research Asia
Beijing, China

Tanzeem Choudhury
Cornell University
Ithaca, New York, USA

Abstract

The rising popularity of the sensor-equipped smartphone is changing the possible scale and scope of human activity inference. The diversity in user population seen in large user bases can overwhelm conventional one-size-fits-all classication approaches. Although personalized models are better able to handle population diversity, they often require increased effort from the end user during training and are computationally expensive.

In this paper, we propose an activity classification framework that is scalable and can tractably handle an increasing number of users. Scalability is achieved by maintaining distinct groups of similar users during the training process, which makes it possible to account for the differences between users without resorting to training individualized classifiers. The proposed framework keeps user burden low by leveraging crowd-sourced data labels, where simple natural language processing techniques in combination with multi-instance learning are used to handle labeling errors introduced by low-commitment everyday users. Experiment results on a large public dataset demonstrate that the framework can cope with population diversity irrespective of population size.

Introduction

With the explosion of smartphones, it is now possible to collect real-time daily activity data over a large population. This continuous availability of a vast amount of data changes the possibilities of human-centric applications and sensing.

But, as the scope of the system broadens from carefully-controlled experiments to mass-generated data, the conventional computational methods for activity recognition are overwhelmed by user heterogeneity in terms of age, behavioral patterns, lifestyle and so on. Performance degradation of classifiers in activity recognition due to the difference between people is known as *population diversity problem*. It has been shown (Lane et al. 2011b) that the population diversity problem can seriously affect the classification accuracy even when the population consists of as little as fifty users.

While personalized models (Longstaff, Reddy, and Estrin 2010; Stikic and Schiele 2009) usually fare much better in handling population diversity, the improvement comes at the expense of increased user involvement. The accuracy of classification requires each user to provide carefully labeled data-segments. As the classifier works in isolation, it leads to redundant efforts while learning about the same activities over similar users.

To handle population-diversity in a practical way, a number of studies (Lane et al. 2011a; 2011b) suggested networked approaches by sharing data across users. To ensure reasonable accuracy, crowd-sourced training samples are weighted according to forms of inter-personal similarity. But, none of the proposed methods scale well with an increasing population. For a large user-base, the cost of i) computing pair-wise similarity network and more importantly, ii) exponential cost of training classifiers with huge datasets resulting from crowd-sourcing, gets impractical even with tens of users (Lane et al. 2011b). As inter-personal differences is more of an issue for a large user base, being not scalable severely limit the usability of existing approaches.

This paper proposes a novel scheme to handle population diversity in a scalable way. We maintain groups of similar users to bring down the cost of computing similarity networks and using the similarity measures for training. To achieve comparable accuracy while keeping the user-burden low, our framework focuses on ensuring exploiting crowd-sourcing within a group. To enable robust crowd-sourcing even in the case of unreliably labeled data from users, we handle the two common errors (Peebles et al. 2010) — semantic discrepancy in the labels and the overlapping boundary of activities. To handle semantic discrepancy, we propose to consider it as a Natural Language Processing (NLP) problem. And, to handle overlapping class boundary resulting from inaccurate start and ending times, we use Multi-Instance learning.

The contributions of the paper are:

- The proposed framework handles population diversity in a way that remains practical even as the user population increases in size. To do so, we maintain group of similar users and limit embedding inter-personal similarity within the group.

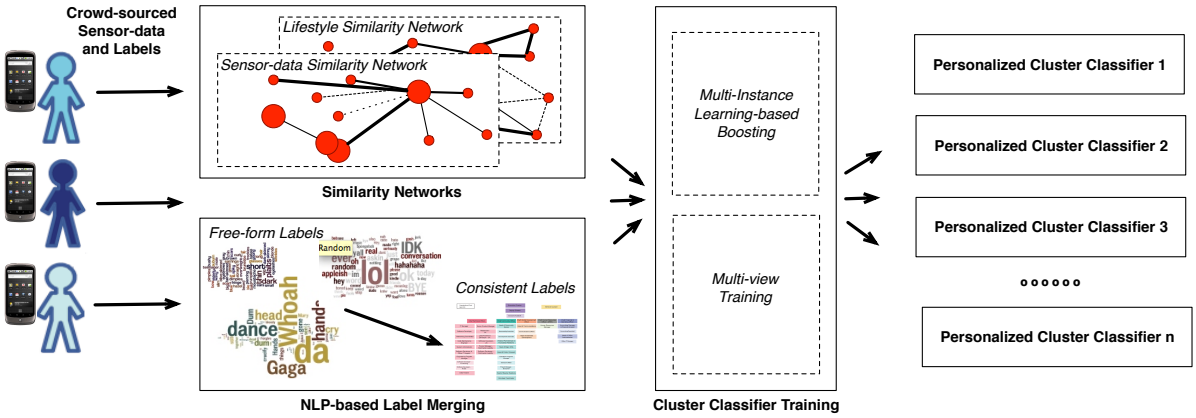

Figure 1: The processing phases of the proposed activity recognition framework for handling data diversity and labeling inconsistencies.

- We enable more robust crowd-sourcing. Previous work on sharing training activity data across users assumes that the labels are consistent. This assumption might be true in case of conventional controlled environments, but while working with a large population of low-commitment users, it is necessary to be robust enough against inconsistent labels.
- By using a large public dataset, we evaluate our framework, both in terms of accuracy and scalability.

Framework

Figure 1 shows the different steps in our framework for producing personalized classifiers that can cope with the population diversity problem in a scalable manner. We describe the steps in more details below.

Similarity Network

As the population grows, the user base starts to get more diverse. Apart from visible demographic dissimilarities like age, weight, gender or fitness level, the population starts to get more diverse in terms of behavioral and lifestyle pattern. As a result, even the core activities like walking can have different signature in sensor data across different group of people. For example, in CSN (Lane et al. 2011b) the authors pointed out the difference in features for two distinct subgroups of users performing walking as seen in Figure 2.

As this inter-personal dissimilarities manifests as the differences in the pattern of raw data, previous approaches (Lane et al. 2011b; 2011a) use similarity networks for training classifiers. In a similarity network graph, each node represents a user and the edge-weight represents similarity between two users. There can be multiple similarity networks to leverage affinity among users in different dimensions — physical similarity network might be used to recognize running while diurnal patterns might be leveraged while inferencing commuting activities. Each classifier for every user is trained on a dataset consisting of weighted data samples from all the users in the population based on the similarity networks. CSN (Lane et al. 2011b), for example, maintains three different similarity networks and a different classifier is trained for each network. To incorporate user similarity while training a classifier for user u_i, the data samples from other user u_j is weighted according to their similarity, $\mathbf{S}(u_i, u_j)$ at the initial iteration. So, for each user u_j in the population, data sample x_{u_j} from that user has the initial weight as

$$\text{weight}^{(0)}(x_{u_j}) = \mathbf{S}(u_i, u_j).$$

As a result, the computational cost of training classifiers can grow out of hand even with tens of users.

To make training of classifiers feasible over a large user-base, we propose to cluster similar users and constrain the crowd-sourcing of data to only users within the same cluster. So, for a fixed number of clusters the number of classifiers trained remains constant irrespective of any increase to the size of the user population. For each type of similarity networks, we use different sets of clusters to leverage dif-

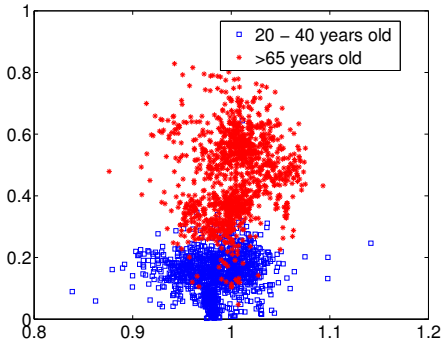

Figure 2: The difference in accelerometer features as two distinct subgroups perform the same activity — walking (originally published in Lane et al. 2011b). The first two principal components of the features are shown above.

ferent dimension of affinity among users. Here we describe clustering users depending on two different notion of similarity — sensor-data similarity and lifestyle similarity. But, it should be noted that other affinity metrics can easily be accommodated in the framework.

Sensor Data Similarity As shown in Figure 2 the difference among users can manifest as the difference in raw sensor-data. So, the similarity in the accumulated data between two users can be a good indicator of inter-personal affinity. Given two users u_i, u_j and the corresponding accumulated feature sets F_{u_i} and F_{u_j}, the similarity function can be defined as the overlap between sets, $\mathbf{S}(u_i, u_j) = \frac{|F_{u_i} \cap F_{u_j}|}{|F_{u_i} \cup F_{u_j}|}$, known as the Jaccard coefficient. But, computing the similarity metric across a huge activity dataset for a large user population is clearly not feasible. So, we use sublinear time near-neighbor search known as Locality Sensitive Hashing (LSH) (Indyk and Motwani 1998).

LSH is a well known technique (Buhler 2001; Ravichandran, Pantel, and Hovy 2005; Das et al. 2007) to efficiently find near-neighbors in a large database. In LSH, data points are hashed using multiple hash functions so that collisions between similar points occurs with higher probability. Finding near-neighbors requires hashing the query point as well to locate the buckets it belongs to. For the Jaccard coefficient similarity, there exists a LSH scheme called Min-Hashing (Cohen 1997).

To use Min-Hashing, we need to randomly permutate the set of all feature S vectors and the hash value for each user u_i is the index of the first feature vector in the permutated set that belongs to F_{u_i} — the set of feature vector for user u_i. For this random permutation, uniformly chosen over the set of all permutations of S, the probability of collision is exactly same as the Jaccard coefficient (Cohen 1997; Broder 1997; Cohen et al. 2001). The Min-Hash produces a set of hash buckets where the probability of two users u_i, u_j being in the same bucket is same to $\mathbf{S}(u_i, u_j)$ — essentially working as a probabilistic clustering algorithm where each bucket is a cluster.

To ensure higher precision in clustering, we can concatenate p hash-values (Indyk and Motwani 1998) so that the probability of two users being in the same bucket is $\mathbf{S}(u_i, u_j)^p$, for $p > 0$. To avoid low recall resulting from the clusters being too refined, we repeat the steps q times. Each user belongs to q clusters where each cluster is defined by concatenation of p hash values.

Permutating the set of feature vectors for the whole activity dataset is computationally unfeasible. Instead, we generate $p \times q$ independent, random hash-seeds and each feature vector is mapped to a corresponding hash-value. The hash-values then serves as the index in the permuted set — resulting in having similar characteristics to the ideal Min-Hash (Indyk 1999).

Lifestyle Similarity The diversity in lifestyle (as measured by location and temporal patterns) can provide an important insight into the context of different activities. Depending on diurnal patterns and mobility distribution same activities can have different signature. The use of lifestyle similarity like diurnal patterns, has been shown to be beneficial in inferring different activity classes such as driving (Lane et al. 2011b).

In CSN (Lane et al. 2011b), lifestyle similarity has been computed from mobility patterns and diurnal patterns by tessellating data into m distinct bins. For GPS estimates in mobility patterns, the bins can be two dimensional and for diurnal patterns each bin can represent the hour during a day in the week — ranging from 0 to 167, 0 denotes the start of the week while 167 marks the final hour of the last day. For each user, CSN would construct a histogram $\{T^{(k)}, k \in [1, m]\}$ of these bins. Histogram frequencies are normalized and the value of the histogram vector reflects the distribution of the data belonging to the user. CSN defines the lifestyle similarity between two users u_i, u_j as $\sum_{k=1}^{m} T_{u_i}^{(k)} T_{u_i}^{(k)}$.

Given the relatively low dimension of the histogram vectors, the above similarity measure can be used in common clustering algorithms. But, for large user-base, we suggest using *Earth Mover's Distance* (EMD). Given two different lifestyle histogram $T^{(k)}$ and $T^{(m)}$, the earth mover's distance EMD($T^{(k)}, T^{(m)}$) is defined as the minimum cost of transforming one distribution to other. This is a popular metric in image and computer vision research (Rubner, Tomasi, and Guibas 2000; Zhao, Yang, and Tao 2010). It is expensive to compute as exact distance requires a solution to minimum transportation problem (Hitchcock 1941). As a result there has been extensive work in approximating the distance metric efficiently (Ling and Okada 2007; Shirdhonkar and Jacobs 2008). More importantly, (Charikar 2002) has shown that LSH scheme exists for EMD.

Robust crowd-sourcing

By enabling crowd sourcing, classifiers can use the steady stream of data from other users to find more discriminating examples to be incorporated into the model. But, given that one of the major goal is to keep the user burden low, the discrepancy in the labels provided by the users with low-commitment is unavoidable. For our framework to be robust enough against labeling errors and inconsistencies, we specifically focus on semantic discrepancy and boundary overlapping.

Semantic Discrepancy in Labels Prior work in activity recognition usually makes the assumption that labels are consistent and the label domain remains fixed. While this might be true in simple scenarios, when people can enter free-form text as label to mark activities, the issue of assigning different labels to similar activities starts to become a concern (Peebles et al. 2010). Similar activity with different labels dilutes the training pool and essentially confuses the classifier.

Given the textual nature of the labels, we suggest to consider finding similar activities as a NLP problem. Specifically, we use a similarity measure in terms of semantic distance between class labels to find similar classes. The semantic distance can be calculated from hyponyms constructed from WordNet hierarchy (Fergus et al. 2010). After finding similar labels, we merge the samples under a generic single label if the number of training samples fall below an

experimentally determined threshold. Otherwise, the data samples are shared during training weighted by their similarity.

Boundary overlapping Data collection for activity on mobile phone usually requires users to mark the start and end of the activity. This can lead to recall errors, lack of temporal precision and interruptions. For example someone labeling gym may forget to mark end point resulting in overlapping boundary with driving data, which can affect the performance of classifiers. Multi-Instance Learning (MIL) can handle boundary overlapping robustly because of the more flexible labeling assumptions.

In MIL, the samples are not treated as positive or negative — labels are assigned to a set of instances grouped together into "bags". For a bag i and sample j in the bag, the probability of a single instance being positive is denoted by p_{ij}.

We adopt the Noisy OR model for each bag. The probability that a bag is positive is given by

$$p_i = 1 - \prod (1 - p_{ij}) .$$

This essentially means that a bag is labeled positive if it contains at least one positive sample, otherwise, it is labeled as negative. So, under MIL settings, the bag labels provide only partial information — it needs to cope with the ambiguity of not knowing which of the instances are positive and which ones are not. But, at the same time, the effect of noise is minimized in classifier training.

MIL has been successfully applied to image segmentation (Vezhnevets and Buhmann 2010), face detection (Guillaumin, Verbeek, and Schmid 2010) and handling label noise in video classification (Leung, Song, and Zhang 2011). It has also been used in activity recognition in sparsely labeled data (Stikic and Schiele 2009). We use boosting based MIL where all instances contribute equally and independently to a bag's label (Xu and Frank 2004).

Multi-view of data As data collection from smartphones is transparent to the user, a large pool of unlabeled data can quickly accumulate. To make use of this plentiful and otherwise wasted data, we suggest using unlabeled data to augment the classifier model.

When multiple similarity networks are available, similar to CSN (Lane et al. 2011b), we suggest to exploit multiple views of different classifier by using multi-training (Blum and Mitchell 1998). But, if there is only one similarity network available, En-Co-Training (Guan et al. 2007) or democratic co-learning (Zhou and Goldman 2004) can be used as they do not make assumptions about independent views of the data.

In this approach, each classifier keeps track of labeled and unlabeled crowd-sourced data and iteratively tries to label the unlabeled data of other classifier. After each such iteration, classifiers are retrained by using the additional new labels as assigned by other classifiers.

So, the steps in our framework can be summarized as:

- Cluster similar users into a group. Similarity networks are formed within each of these clusters.
- Finding semantically similar textual labels and reassign labels.
- Training a MIL inspired Boosting algorithm to learn activities from the shared training samples in a group where each sample is initially weighted by user similarity.
- Using multiple views of the data for leveraging the large amounts of unlabeled data that is crowd-sourced.

Evaluation

In this section, we evaluate the effectiveness and justify our design choices. The following experiments show that the framework scales much better than previous methods without sacrificing accuracy.

Dataset

For evaluation we use a large public dataset from ALKAN system (Hattori et al. 2010). This dataset contains data from more than 200 users and consists of over 35,000 activities. The data was gathered from the mobile device clients — from iOS and Android applications. The dataset contains three axis accelerometer data from daily, real-life movement for more than a year resulting in relatively large dataset that can provide a good insight about the probable scalability issues that might arise in large-scale deployment. This dataset also contains activities with semantically close labels like train.sit, chat.sit, sit and so on.

Some records in the dataset has inconsistent number of data-samples in terms of activity duration. We think the inconsistency arises when the phone can not sample sensor reading at the specified sampling rate, e.g., when talking on the phone. To identify errors in duration, we performed a time-window based sanity check by using the time-stamps in the data. The idea is, assuming that data is sampled at 20Hz, a chunk of data containing N consecutive samples represents a time-window of $\frac{N}{20}$ second. So, reading every N samples and comparing the values in the time-stamp column will give an insight into the variance present in that window. Around 1.1% of total data-sample was discarded because of inconsistency in time-stamp.

Feature Computation

For feature computation, a window size of 128 samples is used with 64 samples overlapping between consecutive windows. For the sampling rate of 20Hz, each window represents 6.7 seconds of data. Mean, energy, frequency-domain entropy and correlation features were extracted from the sliding windows signal.

The DC feature in the sample window is the mean-acceleration value of the acceleration. The energy feature is a normalized sum of the squared discrete FFT component magnitudes of the signal excluding the DC component. Frequency-domain entropy is calculated as the normalized information entropy of the FFT magnitudes — DC feature is excluded from the calculation. Correlation is calculated between all pairwise combination of axes.

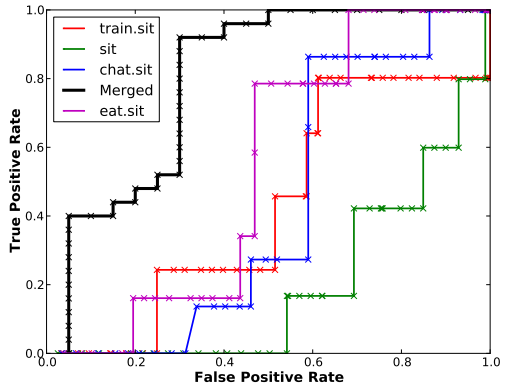

Figure 3: This ROC curve illustrates the effect of merging semantically similar labels. The classifier accuracy using the merged labels outperforms classifiers trained using only the original labels provided by users.

Effect of Merging Labels For evaluating the effect of merging labels, we consider the activities associated with sitting and walking. The activity "sitting" consists of the labels train.sit, chat.sit, sit and eat.sit and for walking the labels are walk.slow, escalator.walk.up, escalator.walk.down and walk. These activities have been selected because significant amount of data have been recorded for each label. The dataset contained more than 46 hours of activities. For each label related with sitting we train a classifier where all other activities are marked as negative samples. For merged labels, we train the classifier with activities related with sitting marked as positive examples having different weights and walk related activities are marked as negative examples. For performance measurement, we use ten fold cross-validation.

From figure 3, the performance gain in the classifier trained from merged label is apparent from the top-left-most placement of the ROC curve. The accuracy of the classifiers trained by isolated labels are rather poor, but it is consistent with earlier findings (Hattori et al. 2010).

Robustness against boundary overlapping In activity recognition systems obtaining high-quality training data has always been a central issues. For large-scale deployment, the problem is more severe. Recording sensor data through real-life, daily activities means lack of temporal precision and frequent disruption. To study the effect of such noise we switch some negative samples to positive samples — simulating the interruption by activities in the middle of recording. We create dataset with $1\%, 5\%, 10\%$ and 20% noise in positive labels. We train a MIL AdaBoost and a simple AdaBoost classifier using same dataset. In both cases, the weak classifier is a C4.5 decision tree. The result is shown in Figure 4. It is apparent that MIL based methods performs much better in the presence of noisy data.

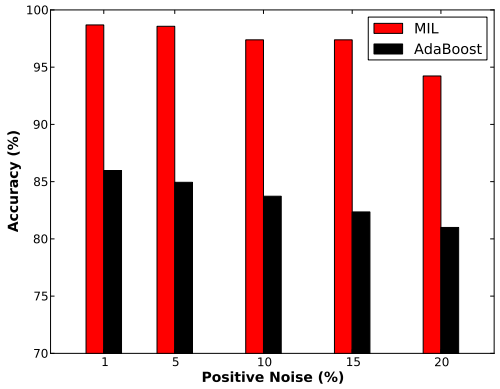

Figure 4: Performance of multi-instance learning in handling activity labels with overlapping boundaries.

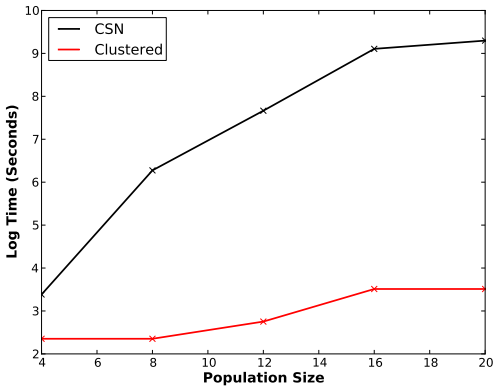

Figure 5: Time (in seconds) to train a classifier for a single user.

Scalability The cost of training classifiers is the bottleneck of deploying activity recognition system that uses similarity networks. To evaluate how well our system scales with an increasing population, we select twenty users and train classifiers to recognize activities having labels sit, walk and stand consisting of more than 686 hours of sensor data.

We compare the training time of classifiers with CSN (Lane et al. 2011b) which uses a fully connected weighted-graph for learning models. While CSN used a computer cluster for training, in our evaluation we use a single machine (2.3 GHz Intel Core i5 CPU and 4 GB of memory). We limited the evaluation to twenty users and single type of similarity network since CSN would take too long to train otherwise. Figure 5 shows the effect of increasing the population while training a classifier for a user based on lifestyle similarity network. The training time of a classifier in our framework is constant for a fixed number of clusters. Consequently, increasing the population does not incur much additional cost. In contrast, the training time for CSN, from 4 users to 20 users increases by 370 times, making it imprac-

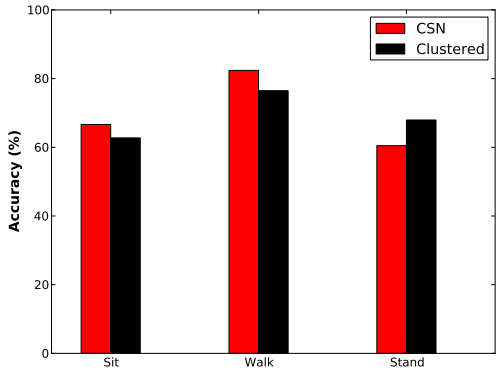

Figure 6: Classifier accuracy in a population of 20 users.

tical to use in large user base.

The important question is how our clustering approach to manage computational cost will affect classifier accuracy. For measuring classifier performance we use a separate test set containing around three hours of sensor data for sit, walk and stand. Figure 6 shows the accuracy of the classifiers for this experiment. The classifier for CSN has been trained on the fully connected lifestyle similarity graph for twenty people while the clustered classifier has been trained on five users with high lifestyle similarity. We selected these three activities and limit the dataset to a single similarity network of twenty users because of the computational cost associated with training for CSN. From the result, we can say that training using clustered users maintains reasonable accuracy while keeping computational cost low.

Conclusion

In this paper, we introduced a scalable way to handle the population-diversity problem. We demonstrated that our framework scales well as the user population increases without sacrificing classification accuracy. Furthermore, our framework introduces new techniques for coping with crowd-sourced labeled activity data, which although can be plentiful can also be prone to error. Our results showed the effect in classifier accuracy due to user disagreement with activity class semantics (e.g., labeling the same activity class with different textual descriptions). We demonstrated how this problem can be improved with NLP-based techniques proposed in our framework. Finally, we introduced techniques specifically to handle segmentation errors during crowd-sourcing, which occur when users make mistakes as to precisely when activities start and end.

While the results are promising, there are still challenges related to long-term usage by a large user population. Like the conventional methods, our framework assumes complete knowledge of the similarity network between all pair of users. This will not be available, for instance, as new users join the system for whom there will be insufficient data to compute a similarity network. Additionally, we assume these similarity networks are static, which ignores the significant drift in user behavior that will occur over time. We plan to further work on the framework to address these design and implementation issues with the eventual goal of coming up with a framework which can be deployed over large population.

References

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 92–100. ACM.

Broder, A. 1997. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences.*, 21–29. IEEE.

Buhler, J. 2001. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics* 17(5):419–428.

Charikar, M. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 380–388. ACM.

Cohen, E.; Datar, M.; Fujiwara, S.; Gionis, A.; Indyk, P.; Motwani, R.; Ullman, J.; and Yang, C. 2001. Finding interesting associations without support pruning. *IEEE Transactions on Knowledge and Data Engineering* 13(1):64–78.

Cohen, E. 1997. Size-estimation framework with applications to transitive closure and reachability. *Journal of Computer and System Sciences* 55(3):441–453.

Das, A.; Datar, M.; Garg, A.; and Rajaram, S. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, 271–280. ACM.

Fergus, R.; Bernal, H.; Weiss, Y.; and Torralba, A. 2010. Semantic label sharing for learning with many categories. *Computer Vision–ECCV 2010* 762–775.

Guan, D.; Yuan, W.; Lee, Y.; Gavrilov, A.; and Lee, S. 2007. Activity recognition based on semi-supervised learning. In *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007.*, 469–475. IEEE.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2010. Multiple instance metric learning from automatically labeled bags of faces. *Computer Vision–ECCV 2010* 634–647.

Hattori, Y.; Inoue, S.; Masaki, T.; Hirakawa, G.; and Sudo, O. 2010. Gathering large scale human activity information using mobile sensor devices. In *2010 International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, 708–713. IEEE.

Hitchcock, F. 1941. The distribution of a product from several sources to numerous localities. *Journal of mathematics and physics* 20(2):224–230.

Indyk, P., and Motwani, R. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality.

In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 604–613. ACM.

Indyk, P. 1999. A small approximately min-wise independent family of hash functions. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, 454–456. Society for Industrial and Applied Mathematics.

Lane, N.; Xu, Y.; Lu, H.; Campbell, A.; Choudhury, T.; and Eisenman, S. 2011a. Exploiting social networks for large-scale human behavior modeling. *Pervasive Computing, IEEE* 10(4):45–53.

Lane, N.; Xu, Y.; Lu, H.; Hu, S.; Choudhury, T.; Campbell, A.; and Zhao, F. 2011b. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the 13th international conference on Ubiquitous computing*, 355–364. ACM.

Leung, T.; Song, Y.; and Zhang, J. 2011. Handling label noise in video classification via multiple instance learning. In *2011 IEEE International Conference on Computer Vision (ICCV)*, 2056–2063. IEEE.

Ling, H., and Okada, K. 2007. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(5):840–853.

Longstaff, B.; Reddy, S.; and Estrin, D. 2010. Improving activity classification for health applications on mobile devices using active and semi-supervised learning. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 1–7. IEEE.

Peebles, D.; Lu, H.; Lane, N.; Choudhury, T.; and Campbell, A. 2010. Community-guided learning: Exploiting mobile sensor users to model human behavior. In *Proc. of 24th AAAI Conference on Artificial Intelligence*, 1600–1606.

Ravichandran, D.; Pantel, P.; and Hovy, E. 2005. Randomized algorithms and nlp: using locality sensitive hash function for high speed noun clustering. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 622–629. Association for Computational Linguistics.

Rubner, Y.; Tomasi, C.; and Guibas, L. 2000. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* 40(2):99–121.

Shirdhonkar, S., and Jacobs, D. 2008. Approximate earth movers distance in linear time. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. (CVPR)*, 1–8. IEEE.

Stikic, M., and Schiele, B. 2009. Activity recognition from sparsely labeled data using multi-instance learning. In *Proceedings of the 4th International Symposium on Location and Context Awareness*, LoCA '09, 156–173. Berlin, Heidelberg: Springer-Verlag.

Vezhnevets, A., and Buhmann, J. 2010. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3249–3256. IEEE.

Xu, X., and Frank, E. 2004. Logistic regression and boosting for labeled bags of instances. *Advances in Knowledge Discovery and Data Mining* 272–281.

Zhao, Q.; Yang, Z.; and Tao, H. 2010. Differential earth mover's distance with its applications to visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(2):274–287.

Zhou, Y., and Goldman, S. 2004. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 594–602. IEEE.