[[1]](#footnote-1)

Dirty jobs: the reality of working with wearable activity tracker data

Marcus D. Sherman

*Abstract*— Obesity is a disease that is a major contributor cardiac disease, type 2 diabetes, and general health and well-being. Therefore, it is important for us to understand how to successfully battle the disease. Using only triaxial armband accelerometer data collected by an ongoing comprehensive multi-phase weight-loss study, this study attempted to build a machine learning classifier that could predict the success of consistent weight-loss. Best practices of cross-validation and data partitioning were implemented to optimize and train a series of machine learning models (gradient boosted tree classifier, random forest, support vector machine, multi-layer perceptron, K-nearest neighbor, & logistic regression) on labeled data. Other ensemble methods such as bagging, stacking, and voting classifiers were also explored. The outcome produce highly unstable classifiers that could not reliably predict better than chance (50%). The instability was in part due to lack of a large enough dataset and to the nature of the disease itself. This study shows that consistent weight-loss is contingent on more than just physical activity alone.

*Index Terms*—machine learning, ensemble learning, accelerometer, supervised learning, obesity

# INTRODUCTION

W

hile obesity continues to be an epidemic in the United States, little research has shown exactly the components that play into a subject’s ability or capacity to lose weight. As individuals increase their physical activity; their cardiometabolic risk profiles improve, they become more resistance to non-communicable diseases, they are less likely to succumb to type II diabetes, and generally live longer lives [5]. To properly test the influence of physical activity regarding weight loss, large-scale studies generally rely on the self-reporting of the participants. Herein lies a bias regarding the disease.

As self-reporting tends to be more reliable when physical activity is more vigorous, light activity is often poorly reported [5]. However, participants in the study are unlikely to undergo continuous vigorous workout regimens (e.g. High intensity interval training, HIIT) regularly as this leads to early burnout. Successful weight-loss often starts with appropriate scaling of physical activity. This often means starting with light physical activity will both stimulate weight-loss and support broader reaching lifestyle changes.

To objectively and accurately capture data on light activity levels and sedentary behavior, accelerometers are often used. An accelerometer is a small, lightweight device that can be worn on several different positions of the body. Conceptually, all accelerometers use an axial spring mass mechanism [6]. As acceleration occurs on a given axis, the force of the acceleration is applied to a mass attached to a “spring”. The spring force is then used as a proxy of acceleration. Ideally, an accelerometer provides outputs of body movement acceleration, acceleration of gravity, and the acceleration of outside forces (such as driving) [1].

Using a triaxial (vertical, mediolateral, anteroposterior) accelerometer, clinicians can monitor the physical activity level of subject without direct observation, little cost, and little discomfort to the subject. Furthermore, since the device is designed to be wearable over time, accelerometers can capture both the intensity of movement and frequency of activity the subject [6].

There are some caveats to using such devices though. The data is inherently noisy due to vibrations of soft-tissue (muscle twitching or leg shaking), external vibration (such as riding in a car), and subject to bias due to device placement. If the device is worn on the waist, it might not capture the intensity of heel strikes or upper body movement [1]. If worn on the ankle, the acceleration of the ankle during a run may disproportionately influence the readings. The same can be said with the wrist and arm. Furthermore, accelerometers cannot discern differences in acceleration due to changes in posture or load (weight) working upon the subject [5].

Since the data collected from accelerometers was continuous time-domain measurements, it stood to reason that contemporary signal processing and machine learning could be applied to discern the differences between successful weight-loss study participants and failures. Furthermore, identified distinctions could then be used to supplement current federal guidelines for physical activity regarding health and fitness and shore the gap of knowledge with respect to light physical activity.

# Methods

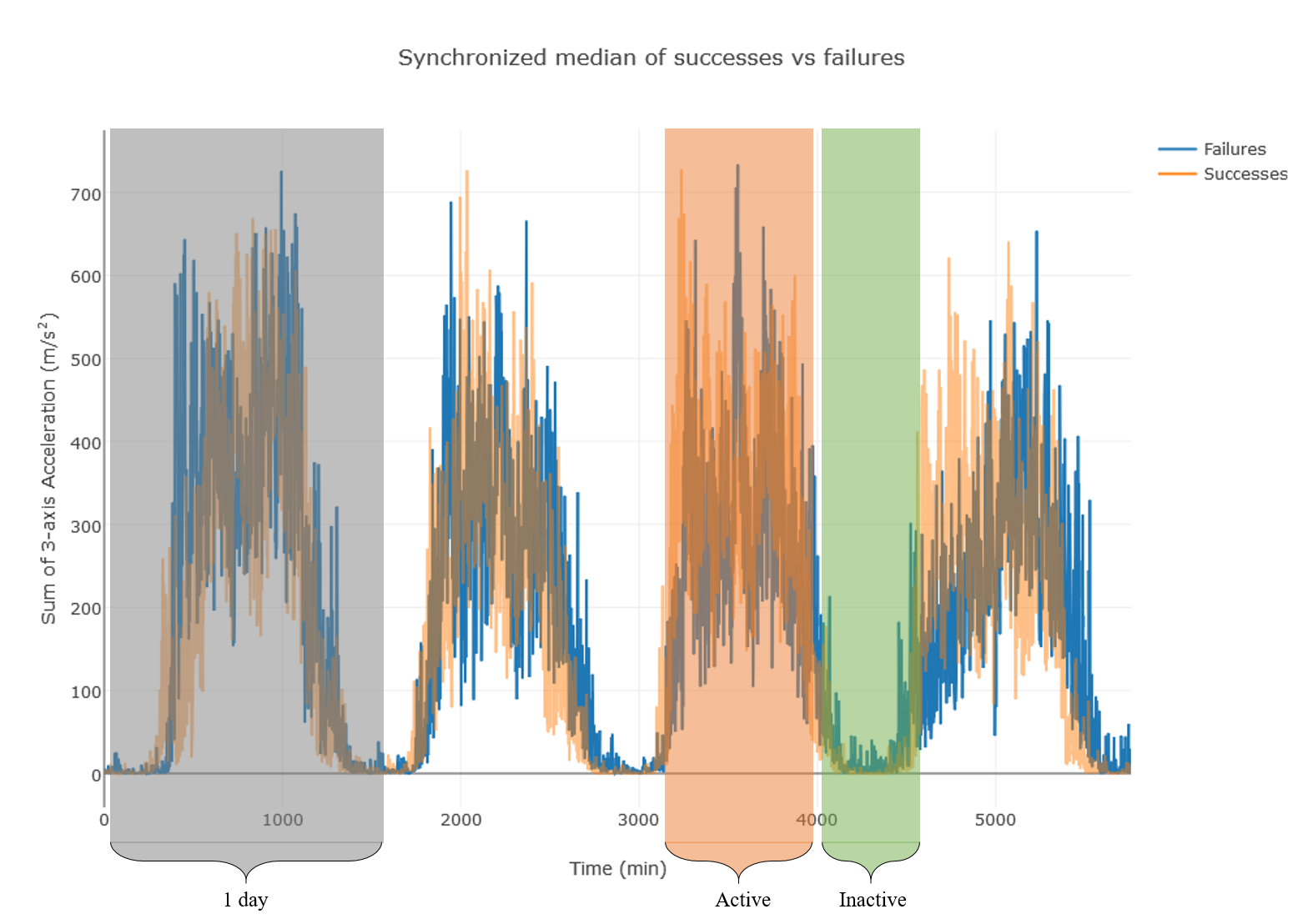
The data used in this study comes from triaxial armband accelerometer data of 100 samples in an ongoing multi-phase weight-lost study. Each sample has 7000 data points representing one reading per minute over the course of five days. 21 of the subjects were labeled as successes after phase 2 of the study. A success label was derived from both reaching a specific weight-loss percentage threshold in phase 1 as well as reaching a similar threshold in phase 2. For the purposes of this research, we only considered the armband data to determine the dependence of physical activity on weight-loss.

Fig. 1. Synchronized median of successes (orange) vs failures (blue). The y-axis is the sum value of the triaxial accelerometer. The x-axis is length of time measured in minutes. Notice the periodic trend of active and inactive periods over the course of 4 full days (5760 minutes). The grey bar depicts a single 1-day segment (4 total) of the data used in feature generation. The orange bar depicts one of the 4 active periods also used in feature generation. Lastly, the green bar depicts one of 3 full inactive periods used in features selection.

The data encompasses five days of free-living subject activity level, each of the samples were synchronized to a complete 4-day cycle (i.e. coupled sleep-wake rhythm). Since the data was unbalanced between failures and successes, an even data set was produced by taking out all 21 successes and randomly sampling 21 of the failures. When both group’s median readings were plotted, it appeared as if successes were more active during the day and less active during the night (Fig. 1).

As activity level is integral to the analysis, the signal was synchronized on their first long inactive period. The data was also analyzed based on the day/night cycle of the subject. Furthermore, activity cut points were used to further discriminate the data. A cut point is defined as identifiable period of the data that marks the difference between active vs inactive periods. As the data was naturally coarse, the highest resolution of these cut points are only periods of long active periods (subject is awake) and long inactive periods (subject is asleep). Fig. 1 depicts the different segments of the data used in feature generation.

Table 1 shows the features that were generated for the complete signal, each full day, and each active and inactive period observed in the data. Furthermore, the differences between each of the days (to detect trend of activity) and sleep vs wake period were calculated. In total, 336 features were generated

As overfitting is always an issue with any machine learning algorithm, the dataset was partitioned for each of the following steps in a stratified fashion: Feature selection, training, and model testing. Fig. 2 shows the partition workflow for the project.

Feature selection was done by selecting the 21 best features based on mutual information score [13] and by recursive feature elimination & K-fold cross validation (RFECV) [4]. RFECV iteratively trains a given model, optimizing the weights a given feature has on the classification based on its receiver operator curve area under the curve score (ROC-AUC), and recursively eliminates the feature with the lowest rank until the smallest number of features exist with the highest accuracy the model can achieve is left. Ultimately, K-best feature selection based on mutual information was used for all other downstream analysis. See discussion for more information regarding this choice.

First, the following hyperparameters of a gradient boosted decision tree (GBT) classifier [2] were optimized via grid search with 6-fold cross-validation: learning rate, number of base estimators, max tree depth, minimum samples to split on, minimum samples per leaf, maximum leaf nodes, and subsample size. A deviance (logistic regression) loss function was used for classification purposes. The GBT algorithm is an ensemble machine learning method that utilizes multiple estimators for classification. More information about GBT can be found in section IIIA below.

Additionally, a classification and regression tree (CART) [5] was optimized for many of the same hyperparameters as GBT, except the number of learners. However, an intermediate step between a CART and GBT is a Random Forest (RF) classifier [2]. In the simplest way, a random forest uses multiple smaller CARTs for classification to protect against overfitting. It was optimized by grid search with 6-fold cross-validation for the following hyperparameters: maximum features, maximum depth of tree, minimum samples to split on, minimum samples per leaf, etc.

A simple logistic regression classifier and support vector machine classifier (SVC) was also optimized using a grid search with 6-fold cross-validation. The logistic regression hyperparameters that were tuned were the penalty function (l1 or l2), maximum iterations, and regularization strength (C value). The SVC hyperparameters were the penalty parameter (C value), kernel type (linear, polynomial, or radial basis function), degree (when polynomial was selected), and the kernel coefficient (gamma).

For the sake of completeness, a simple K-nearest neighbor (KNN) and Gaussian naïve Bayes classifiers were also implemented. Furthermore, a log-loss multi-layer perceptron (MLP) using a stochastic gradient descent for learning was trained [6]. Being the most complex of the “simple” classifiers, it too was optimized using 6-fold cross-validated grid search of hyperparameters. Some of these parameters were activation function, number of hidden layers, L2 penalty parameter, learning rate, etc.

TABLE I

Generated Features

|  |  |
| --- | --- |
| Domain | Feature |
| Time | Mean, median, variance [9] |
|  | Skewness & Kurtosis |
|  | Max |
|  | Max-min |
|  | Autocorrelation |
|  | Root mean square |
| *Frequency* | Root mean square |
|  | Power |
|  | Mean Frequency [9] |
|  | Power spectral density [9] |
|  | Spectral flux |
|  | Spectral roll-off |
|  | Spectral centroid |

Given that multiple “simple” classifiers had already been implemented and trained, it seemed appropriate to interrogate more complex ensemble algorithms that build easily upon these existing classifiers.

A bootstrap aggregating (bagging) classifier, much like a random forest, uses a series of base classifiers. However, the choice of the base classifier is up to the user (unlike random forests). The difference is that each instance of a base classifier is trained on a different subsample of the training set. This results in a series of independent classifiers trained on separate portions of the training set. Then, by way of a meta-estimator, produces a classification for a sample based on voting or by averaging of the base classifiers. More information regarding bagging can be found in section IIIB.

Whereas random forests and bagging classifiers require the use of a single type of base classifier, a voting classifier derives its strength from using different types or differentially optimized classifiers as its base estimators. Using the previously optimized base classifiers, a voting classifier was also implemented in this study. It simultaneously trains the base classifiers, then queries each of the classifiers for predicted classifications of a sample and by majority or weighted vote, returns the most “popular” classification.

Lastly, a stacking classifier was implemented to round out the comprehensive exploration of ensemble learning techniques. A stacking classifier, much like a voting classifier, use multiple independently optimized classifiers as its base estimators. The key difference is that the parameter space of each of the base estimators become features for the meta-classifier. Here, instead of a simple averaging or voting process, an additional classifier is trained on the parameters of the base estimators. This meta-classifier then uses both the predictions and parameters of the base estimators to produce its own uniquely informed classification of the input data. More information on the stacking algorithm can be found in section IIIC.

# MATH

## Gradient Boosting [2]

The gradient boosting algorithm is an ensemble learning technique that incorporates multiple decision tree base learners that randomly sample the feature space at optimized tree depths. The ranking of the base learners is based on the gradient descent of the tree. Using the most informative base learners, the algorithm builds a “meta classifier” the increase its robustness and accuracy. A description of the algorithm can be found below.

Given a training data set , a specified loss function , and a chosen base-learner where refers to the explanatory input variables, refers to the corresponding class label, and is a parameter estimate. Iteratively computer the negative gradient along the observed data:

(1)

and fits a new base-learner , and finds the best gradient descent step-size :

(2)

and updates the function estimate:

(3)

## Bootstrap Aggregating

Bootstrap aggregating (bagging) is an ensemble algorithm that uses multiple base learners trained on many random subsamples of the training set to create a robust classifier. A description of the classifier can be found below.

Given a training data set where x refers to the explanatory input variables, refers to the corresponding class labels, a sequence of random samples are used such that a classifier function = y trained on the bootstrap aggregate where the estimated class is based on the plurality in of bootstraps.

## Stacking

Stacking algorithms are another ensemble learning technique that uses a series of different types of base learners (e.g. random forest, CART, naïve Bayes, etc.) or differentially optimized base learners for classification. A chosen base learner is selected as a meta learner. It takes the parameters of the trained base learners as features for its overall classification. A description of the stacking method can be found below.

Given a training data set where x refers to the explanatory input variables, refers to the corresponding class labels, and first-level learners where is the number of first-level learners. where and is the trained learner. Then is used to classify a training example. Each classification of each first-level learner is added into a new dataset . The meta learner is then trained on . The resulting classification is .

# Results

Using a small, unbalanced, and noisy dataset of triaxial accelerometer data spanning five days, multiple machine learning models were optimized and trained to classify samples as success or failure. After stratified cross-validation over many iterations, the base classifiers were unable to satisfactorily classify the samples. However, marginal improvements in ROC-AUC for some of the more involved ensemble techniques (except GBT).

The ROC curves of the “simple” classifiers (logistic regression, SVC, CART, and KNN) can be found in Fig. 3A. Logistic regression achieved an AUC of around 0.56. SVC had an AUC of 0.5. CART received an AUC of 0.54. Finally, KNN had an AUC of 0.58.

The ROC curves for the “intermediate” classifiers (GBT, RFECV, and RF) can be found in Fig. 3B. GBT resulted in an AUC of 0.46. Similarly, RFECV (since it was trained using a base GBT classifier) had an AUC of 0.49. Lastly, RF ended with one of the best AUCs of 0.67.

Finally, the ROC curves for the most “complex” classifiers (Voter, bagging, and stacking) can be found in Fig. 3C. Voter and stacker classifiers resulted in AUCs of 0.5, while bagging was a marginally better with an AUC of 0.56.

Furthermore, all the classifier’s AUCs were wildly erratic based on the random partitioning. Despite optimization, all the models were highly unstable. This resulted in classification being around chance or slightly better overall. A sample of the ROC plots can be seen in Fig. 3. Ultimately, no single classifier could effectively and/or appreciably predict success of weight-loss based on armband data alone.

# Discussion

Early forms of data preprocessing including filtering the data through a low-pass third-order Butterworth filter between 0 and 20 Hz. This was due to most human movements being adequately captured within this range [6][9]. As the arm acts a s mass damper to minimize head pitch in the sagittal plane, it acts as a proxy measure of whole body activity. However, due to the low sampling rate (1 per minute), this resulted in a less informative signal and was ultimately left out.

Due to the sparsity of the data and the coarseness of the signal, data reduction was not implemented (outside of initial Butterworth filtering mentioned above). However, data reduction of accelerometer data is a widely studied field and there have been multiple validation studies considering data reduction, there are (as of the writing of this paper) no consensus or general convention [6]. In any future iterations of this work, if adequate accelerometer data is collected, this is a serious consideration.

GBT weights weak learners (random decision trees that have classification accuracy between 0.5 < x < ~0.9) based on gradient descent. It was believed that due to the small differences between the samples, a series of weak learners would be the best approach to classify successes and failures based on accelerometer data. The scikit-learn implementation of the GBT comes out-of-the-box ready for regularization, subsampling, and loss function selection—all of which protect against overfitting [7]. However, GBT requires more samples than available for it to become sufficient in classification. This was something overlooked early on in model selection.

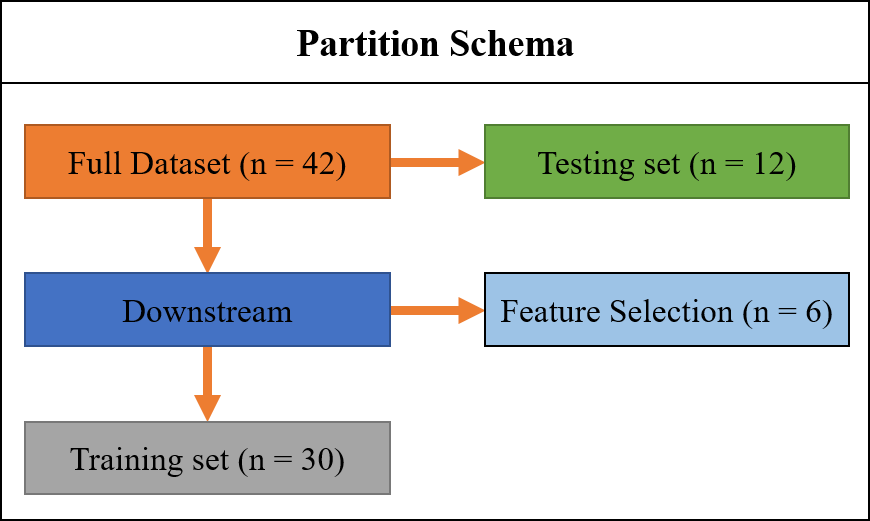


Fig. 2. Partition schema used in the study. After balancing the full dataset with successes and failures (n = 42), a testing set (n = 12) was initially set aside. Further partitioning was done for feature selection (n = 6) and training (n = 30)

RFECV also provided no reprieve despite its promise at simplifying feature selection. The number of features required for RFECV to adequately optimize on greatly outnumbered the number of samples. To protect from overfitting. K-best feature selection was used in its place.

Despite correcting for sample balance and conservative protection against overfitting, no other machine learning model in the scikit-learn arsenal could consistently predict above chance (50%).

# Conclusion

B

C

A

The classification of success or failure of weight-loss based on triaxial accelerometer data could not be sufficiently done with any of the classifiers in the scikit-learn library. There are many potential causes for this result. The first is the sheer lack of sample data. Ideally, more data is always better. In the case of this study, the number important/informative easily outnumbered the number of samples. By reducing the feature space to protect against overfitting, a comprehensive and informative feature space could not be compiled.

The second issue lies in the features themselves. There are potentially better features that were not considered in this study that may have been more informative to the models.

The third issue is the nature of the data itself. Success at weight-loss is more than just exercise. Genetics, diet, and environment all play a major role in weight-loss. It stands to reason that even if the best features were selected, and hundreds of thousands of samples were collected, one’s ability to lose weight may require more than just physical activity. For us to accurately predict ability to lose weight might require additional multi -omic dataset, diet & intake information, heredity & lineage samples, and health & psychological records.

# Future Works

## Interventions could potentially be developed based on early cut point detection in the data (points where the subject transitions to differing levels of activity). These detected cut points and length & degree of physical activity could inform subjects how to better change their activity profiles to encourage efficient weight loss. Additional physiological sensors that measure heart rate, respiration, and muscle contraction may also add another layer of predictive power.

## Providing privileged information such as psychological records and diet information could also help provide a more comprehensive picture of the subject’s progress. Furthermore, a genome-wide association study of the successes may identify some potential biomarkers that could edify the genetic background of what supports weight-loss or genetic changes that occur during a productive weight-loss regimen.

Supplemental Data

Both the Jupyter notebook and raw data can be found at [this](https://github.com/betteridiot/Machine_learning.git) link or this address:

<https://github.com/betteridiot/Machine_learning.git>

References

Fig. 3. Receiver Operator Characteristics (ROC) curves for different subsets of classifiers. The dashed line represents chance (50%) in all plots. A) logistic regression (green) with an AUC of 0.56, SVC (red, obscured by chance) with an AUC of 0.5, CART (purple) with an AUC of 0.54, and KNN (pink) with an AUC of 0.58. B) GBT (blue, obscured by RFECV), with an AUC of 0.46, RFECV (orange) with an AUC of 0.49, and random forest (rf, burgundy) with and AUC of 0.67. C) Voter (yellow-green) with and AUC of 0.5, bagging (light blue) with an AUC of 0.56, and stacking (dark blue) with an AUC of 0.5

1. Bouten, CVC *et al.* (1997). A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE Transactions on Biomedical Engineering*, 44 (3): 136-147
2. Breiman L. (2001). Random Forests. *Machine Learning.* 45(1): 5-32.
3. Friedman JH. (2001). Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 29(5): 1189-1232.
4. Guyon I, Weston J, Barnhill S, Vapnik V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46: 389–422.
5. Hastie T, Tibshirani R, Friedman J. (2009). Elements of statistical learning. Springer.
6. He K, Zhang X, Ren S, Sun J. (2015) Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. ICCV ‘15:1026-1034. doi:10.1109/ICCV.2015.123
7. Lee I, Shiroma EJ. (2014). Using accelerometers to measure physical activity in large-scale epidemiological studies: issues and challenges. *Br J Sports Med.* 48:197–201. doi:10.1136/bjsports-2013-093154
8. Mâsse LC. *et al.* (2005). Accelerometer data reduction: A comparison of four reduction algorithms on select outcome variables. *Medicine & Science In Sports & Exercise.* 0195-9131/05/3711(Suppl)-S544/0 doi: 10.1249/01.mss.0000185674.09066.8a
9. Mathie MJ, Coster ACF, Lovell NH, Celler BG. (2004). Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiol. Meas*. 25: R1–R20. doi: 10.1088/0967-3334/25/2/R01
10. Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. *JMLR* 12: 2825-2830.
11. Preece SJ, Goulermas, JY, Kenney PJ, Howard D. (2009). A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Transactions on Biomedical Engineering*. 56(3):871-879.
12. Pontzer H, Holloway III JH, Raichlen DA, and Lieberman DE. (2009). Control and function of arm swing in human walking and running. *Journal of Experimental Biology* 212: 523-534. doi:10.1242/jeb.024927
13. Ross BC. (2014). Mutual information between discrete and continuous data sets. *PLoS ONE* 9(2): e87357. doi:10.1371/journal.pone.0087357
14. Yu HF, Huang FL, Lin CJ. (2011). Dual coordinate descent methods for logistic regression and maximum entropy models. *Mach Learn*. 85: 41. doi:10.1007/s10994-010-5221-8

1. M. D. Sherman is with the University of Michigan, Department of Computational Medicine and Bioinformatics (mdsherm@umich.edu). [↑](#footnote-ref-1)