Bite Prediction

Ty Lewis, Rob Ivie, Mike Brodie
CS 478, Winter 2014
Department of Computer Science
Brigham Young University

Abstract

There is an abundance of mobile health apps on the market today, and almost all focus entirely on net caloric intake (exercise minus calories consumed). Recording daily caloric intake can be cumbersome and inefficient. One rising suggestion in the health field for reaching weight goals is recording the number of bites during meals throughout the day.

In this paper, we record our efforts to classify bites and non-bites in an effort to automate counting the number of bites during a meal. We first describe our user study and data gathering efforts using a YEI 3-Space Sensor. Next, we analyze our initial results and discuss consequent changes to our trainable dataset. Finally, we discuss our process for improving features and reducing our trainable arff files to only the most relevant input features. We also report on the performance of numerous base-level classifiers such as KNN, Naive Bayes, and decision trees as well as meta-level classifiers (voting, bagging, and boosting) using Weka.

By using Weka, a uniformed window size, mean, standard deviation, covariance, spectral entropy, and energy, our final accuracies ranged between 86% and 100% on almost all learners.

Introduction

Previous research has successfully used accelerometer data to learn to distinguish a host of gestures and body movements. Different mathematical models have been employed–including fast-fourier transform and spectral entropy–to recognize various physical motions. [Lester *et al.*, 2006] Many of these experiments take place in hospitals or elderly care facilities. These learning models allow doctors to safely monitor patients from a distance and receive regular feedback on their physical activities.

Furthermore, these machine learning models allow doctors to link patients' reported pain levels with their corresponding activities. Over time, the machine learning models improve and learn what types of activities are critical or important to report. Although these models can successfully classify activities such as "walking, jogging, [or] riding a bike," previous research has not demonstrated that these models can learn to recognize the motion of taking a bite of food. [Lester *et al.*, 2005]

Our research will extend these projects by training machine learning models to recognize the physical movements of taking a bite. As in previous experiments, we will use "an accelerometer sensor-based approach" to record and derive our data set features. [Khan *et al.*, 2010] Additionally, we will use a wrapper to reduce our data set to the most relevant and predictive features. We expect, much like in Huynh and Schieles research, that this approach will improve "recognition rates" through "careful selection of individual features." [Huynh and Schiele, 2005]

Methods

Data Gathering

With the help of Josh West from the Department of Health Science, we conducted a user study with 13 participants on February 21, 2014. During this study, we recorded 130 instances (bites) of users eating food with a fork. For each participant, we strapped a YEI 3-Space Sensor to the person's wrist that recorded data using a built-in accelerometer, gyroscope, and magnetometer.

Before taking a bite, a participant placed his hand in the start position (the base of the table) and clicked a button on the YEI sensor to indicate the beginning of the measurement time window. After taking a bite and returning his hand to the start position, the participant clicked a second button to mark the end of the bite instance.

In addition to these collected data, we used a data set of 260 bite instances and 91 non-bite instances recorded from a previous user study. In our initial experiments, we believed that this additional data would allow us to train machine learning models to differentiate between bite and non-bite instances.

Initial Data Set

As mentioned earlier, each data instance represented the YEI 3-Space Sensor readings and button readings. To form the trainable arff file from the raw data, features had to be extracted from the gathered data. Other research that has involved machine learning in human movement recognition has used a wide range of varying features. In fact, most studies use as many features as possible. Our initial approach focused on a limited feature set that included:

- $\bar{a}_x, \bar{a}_y, \bar{a}_z$ - The mean acceleration readings for the x, y, and z axes.
- $\sigma_x, \sigma_y, \sigma_z$ - The standard deviation of the acceleration readings for the x, y, and z axes.

It was necessary to analyze these data over fixed time intervals. Finding an appropriate window size that fit the results proved difficult. We estimate that many people eat at different rates, and it may be necessary to create a sub-learner that can learn window size for each individual. However, through manual slicing of our data we were able to find a uniform window that could be used for data extraction.

Data Definition	**Data Instance**
(x,y,z Gyroscope)	(-0.00936,0.01940,-0.00777)
(x,y,z Accelerometer)	(-0.04792,0.97900,-0.05313)
(x,y,z Compass)	(0.24748,-0.43911,0.09929)
Button State	0
...	

Table 1: Explanatory definition of the gathered data

Initial Results

After collecting and transforming the data from our user study into a useable arff file, we ran our data through a series of learning models. Initially, we used backpropagation and decision tree to test the accuracy. Unfortunately, each of these models performed poorly when predicting the output class of an instance (bite or non-bite).

In light of our results, we analyzed our data set for possible errors and ways to improve. During the first iteration, we only included bite instances with data for the x, y, and z means and standard deviations (see Table 2 Iteration #1). In an effort to improve prediction accuracies, we decided to record non-bite instances and add them to our data set. We recorded 89 non-bite instance–bringing the total number of bite instances to 404.

Next, we calculated the x, y, and z covariance for acceleration for each instance in the data set. When we included covariance–as well as the non-bite instances–and ran this dataset through the learning models in Weka, our average accuracy across the models increased from 84.7770% to 86.0032% (See Table 2 Iteration #2).

Feature Improvement

Beyond the temporal domain, we hoped to derive additional features from the frequency domain using the discrete Fourier transform (DFT). These frequency domain features include spectral entropy, energy, and the DC component.

For a bite of length L, the spectral entropy was calculated in two parts. First, a probability density function was calculated over the frequency domain, excluding the DC component:

$$pdf[u] = \frac{\mathcal{F}(b[i])[u]}{\sum_{u=1}^{L/2+1} \mathcal{F}(b[i])[u]}$$

Second, this $pdf[u]$ is used to calculate the signal's entropy e:

$$e = -\sum_{u=1}^{L/2+1} \log_2(pdf[u])pdf[u]$$

Note that here and throughout this paper $\mathcal{F}(\cdot)$ denotes the DFT operator. It is also worth mentioning here that each Fourier domain feature was calculated separately for x, y, and z acceleration readings. (see Table 2 Iteration #3 for the initial results with this feature included.)

The spectral energy ϵ for a bite of length L was calculated using the following formula (again, the DC component is omitted):

$$\epsilon = \sum_{u=1}^{L/2+1} |\mathcal{F}(b[i])[u]|^2$$

Finally, the DC components μ are used as input features as well:

$$\mu = \mathcal{F}(b[i])[0]$$

This feature can be interpreted as the mean energy within the signal.

In total, 18 different features were derived:

1. c_{xy}, c_{xz}, c_{yz} - The covariance for the x and y readings; x and z readings; and y and z readings.
2. $\bar{a}_x, \bar{a}_y, \bar{a}_z$ - The mean acceleration readings for the x, y, and z axes.
3. $\sigma_x, \sigma_y, \sigma_z$ - The standard deviation of the acceleration readings for the x, y, and z axes.
4. e_x, e_y, e_z - Spectral entropy for the x, y, and z acceleration frequency domains.
5. $\epsilon_x, \epsilon_y, \epsilon_z$ - Spectral energy for the x, y, and z acceleration frequency domains.
6. μ_x, μ_y, μ_z - The DC component for the x, y, and z acceleration frequency domains.

Our early data sets–although they yielded high prediction accuracies (see Table 2 Iteration #4, which used all features)–proved to be of relatively poor quality. These initial data sets combine data collected from a smart phone with data recorded using the YEI 3-Space Sensor. Upon later investigation, we discovered that the smart phone orients its axes differently than the YEI 3-Space Sensor. For example, when strapped to the wrist, the android phone points the z-axis upward, perpendicular to the arm; the YEI 3-Space Sensor, however, interprets this direction as the y-axis. Furthermore, the YEI 3-Space Sensor normalizes and corrects its readings for gravity, but the smart phone sensor does not.

In addition to incompatible sensor readings, the window size and sampling rate also emerged as sources of inconsistency. Note that *window size* refers to the number of accelerometer readings in a subsection of the sensor's entire log of readings. For data collected by the smart phone, the windows were selected by hand. But for the YEI 3-Space Sensor, the windows were automatically generated using button presses logged at the start and end of each bite.

Files	Iteration #1 Accuracy (%)	Iteration #2 Accuracy (%)	Iteration #3 Accuracy (%)	Iteration #4 Accuracy (%)
BAYES				
BayesNet	89.1358	88.642	91.1111	94.0741
NaiveBayes	81.9753	82.2222	87.6543	85.9259
NaiveBayesUpdateable	81.9753	82.2222	87.6543	85.9259
FUNCTIONS				
Logistic	84.9383	88.3951	90.8642	93.5802
MultilayerPerceptron	89.6296	95.5556	95.0617	95.5556
SGD	84.4444	87.4074	90.3704	91.358
SGDText	78.0247	78.0247	78.0247	78.0247
SimpleLogistic	84.9383	87.4074	91.1111	93.5802
SMO	79.5062	82.716	85.679	85.9259
VotedPerceptron	77.7778	78.5185	79.2593	49.8765
LAZY				
IBk	91.1111	95.3086	96.7901	96.7901
KStar	90.6173	96.2963	96.7901	95.0617
LWL	81.4815	79.2593	85.9259	90.3704
MISC				
InputMappedClassifier	78.0247	78.0247	78.0247	78.0247
RULES				
DecisionTable	89.8765	89.8765	91.6049	94.0741
JRip	88.8889	90.1235	90.1235	94.0741
OneR	76.2963	76.2963	85.1852	87.9012
PART	90.3704	88.642	94.5679	95.5556
ZeroR	78.0247	78.0247	78.0247	78.0247
TREES				
DecisionStump	78.0247	78.0247	90.3704	90.3704
HoeffdingTree	81.4815	82.716	87.1605	85.679
J48	89.1358	89.3827	95.0617	95.8025
LMT	90.3704	91.358	94.321	94.5679
RandomForest	92.3457	92.3457	96.2963	96.2963
RandomTree	87.1605	89.1358	93.3333	95.0617
REPTree	88.8889	89.3827	92.5926	93.8272
Average Accuracies	84.7770	86.0032	89.3848	90.1010

Table 2: Initial Results

Learning Model	Before Accuracy (%)	With Feature Reduction (%)	With Feat. Reduction and Bagging (%)	With Feat. Reduction and Boosting (%)
Logistic	86.9565	88.4058	89.1304	89.3116
MultilayerPerceptron	86.2319	87.3188	87.1981	87.5
SimpleLogistic	87.6812	88.4058	88.6473	88.7681
JRip	88.4058	85.8696	87.1981	88.0435
PART	86.2319	87.3188	87.6812	87.6812
J48	87.6812	88.0435	88.8889	89.1304
LMT	86.2319	86.2319	86.9565	87.8623
RandomForest	89.8551	90.5797	89.6135	89.4928
Average Accuracy	87.4094	87.7717	88.1643	88.4737

Table 3: Final Results for *uniform-wind-5*

The sensors also differed in the rates at which they captured readings. While the smart phone captured 70 readings-per-second, the YEI 3-Space Sensor only captured 10 readings-per-second. Resultantly, the derived features for the initial instances used windows of data with varying lengths and sampling frequencies.

As shown in Table 3, some of the most salient attributes are the spectral features based on the DFT. The DFT calculation is sensitive to both the size of the window and sampling frequency, producing different results as each attribute varies.

Because of these inconsistencies, we compiled two additional data sets: *uniform_wind* and *uniform_wind_5*. Both data sets contain bite and non-bite instances using a uniform window of 100 readings collected at a uniform rate. In order to derive more consistent attribute values, the data sets only contain positive bite instances recorded using the YEI 3-Space Sensor. The non-bite instances, however, differ in each file. Non-bite instances in *uniform_wind* are derived from data collected by the smart phone on non-bite activities. Although these data were not collected by the YEI 3-Space Sensor, they nevertheless represent arbitrary non-bite gestures. The *uniform_wind_5*'s non-bite instances are derived from the YEI 3-Space Sensor. Many of these instances more closely resemble a bite and require greater precision by the machine learning models in order to be correctly classified. In addition, the non-bite instances include motions such as reading a book, washing one's hands, and reaching for objects.

Feature Reduction Attempts

Although the data set does not have an extremely high dimensionality, we attempted to reduce these 18 features to an optimal subset of features with respect to accuracy. To select the most optimal features, we used the wrapper model. This method works by training and testing a particular model's accuracy using only a subset of the available features. Theoretically, accuracy may be improved by choosing a subset of features that excludes noisy and irrelevant features. Finding such a subset requires searching through a search space of feature subsets; our approach employed Weka's genetic search algorithm to sift through possible subsets.

Bagging and Boosting

With the most predictive features identified using the wrapper, we next tried improving our results further through use of boosting and bagging.

Results

Learning Model	Accuracy (%)
Logistic	100
MultilayerPerceptron	100
SimpleLogistic	99.1597
JRip	95.7983
PART	99.1597
J48	99.1597
LMT	99.1597
RandomForest	100
Average Accuracy	99.0546375

Table 4: Final Results for *uniform-wind*

Tables 3 and 4 summarize our final results for the *uniform_wind* and *uniform_wind_5* data sets. Notice that several of the models achieve 100% accuracy on *uniform_wind*. Because *uniform_wind*'s non-bite instances represent arbitrary gestures, these results suggest that the machine learners are well equipped to distinguish between arbitrary movements compared to actual bite gestures.

Perhaps of greater interest are the results for *uniform_wind_5*. Recall that this data set includes many non-bite instances resembling actual bites. As the accuracies in Table *** suggest, this appears to be a more difficult task. Even after applying feature reduction, bagging, and boosting there is only a slight improvement in accuracy. Despite the decline in accuracy compared to *uniform_wind*, it is remarkable that the machine learners can still achieve near 90% accuracy by each of the models. This is true of all of the tested learning models; no one model significantly outperformed the rest.

Our feature reduction attempts for *uniform_wind_5* also revealed some interesting results. Table 5 shows which features were retained by each model after reduction; theoretically, these features are the most predictive. The right most column of Table 5 indicates the number of models that retained the

Features	Logistic	Multilayer Perceptron	Simple Logistic	JRip	PART	J48	LMT	Random Forest	Retained by # of Models
zSpecEntropy	x	x	x	x	x	x	x	x	8
zstd	x	x	x	x	x		x	x	7
yzCov	x	x		x	x	x	x	x	7
ystd		x	x	x	x		x	x	6
ySpecEntropy	x	x	x				x	x	5
ySignalEnergyMean	x	x		x		x	x		5
zSignalEnergyMean	x			x	x	x	x		5
xzCov		x		x		x	x		4
zSignalEnergy	x		x		x	x			4
xSignalEnergyMean		x		x		x		x	4
xmean	x	x	x						3
ymean	x		x			x			3
zmean	x	x						x	3
xSpecEntropy					x	x	x		3
xstd			x				x		2
ySignalEnergy			x				x		2
xyCov									0
xSignalEnergy									0

Table 5: This table shows which features were retained following feature reduction on each of the models.

Information Gain Ranking	Attributes
0.3608	zSignalEnergy
0.3608	zstd
0.2782	zSpecEntropy
0.2291	xyCov
0.2059	xmean
0.2059	xSignalEnergyMean
0.1523	xzCov
0.0855	zmean
0.0752	zSignalEnergyMean
0	ymean
0	ystd
0	xstd
0	xSignalEnergy
0	ySignalEnergyMean
0	ySignalEnergy
0	yzCov
0	ySpecEntropy
0	xSpecEntropy

Table 6: Attributes ranked by information gain.

feature after reduction; the features have been sorted according to this metric. Features derived from the z and y acceleration readings appear to be the most predictive across the learning models. zSpecEntropy is used by all models. One possible explanation for the predictive quality of features derived from the y acceleration may be because the primary direction of a bite gesture is up and down along the y axis.

Compare these results, however, with those of ranking features according to the information gain each attribute provides (see Table 6). Again, attributes based on z acceleration readings rank higher for predictive ability. This time, however, y's derived features appear much lower. This may be explained by the non-bite instances that do *not* resemble the bite motions. For example, motions such as reaching for objects or washing one's hands provide more motion in the z direction than taking a bite. Thus, initially, a decision tree would likely split based on attributes derived from the z acceleration data to differentiate between bites and non-bites. This may also explain why features derived from the z acceleration feature is retained so often during reduction.

Conclusion and Future Work

Our research reaffirms the possibility of correctly classifying the physical motion of taking a bite. The results from our experiments demonstrate that accuracy steadily improves with the inclusion of calculated attributes such as variance, spectral entropy, and spectral energy. In addition to this, we observed slight accuracy improvements for various machine learning models by bagging and finding an optimal subset of features.

In future experiments, researchers can add more bite and non-bite instances to the data set. A richer data set will allow for better overall learning and improved generalization to new bite instances. These future experiments may also add features to the data set in order to discover the most predictive attributes for classifying bite instances. Finally, future work may branch out to classify a variety of hand-to-head movements–such as scratching the back of the head, touching the nose, or massaging the front of the neck.

References

[Huynh and Schiele, 2005] Tâm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies*, sOc-EUSAI '05, pages 159–163, New York, NY, USA, 2005. ACM.

[Khan *et al.*, 2010] Adil Mehmood Khan, Young-Koo Lee, Sungyoung Y. Lee, and Tae-Seong Kim. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *Trans. Info. Tech. Biomed.*, 14(5):1166–1172, September 2010.

[Lester *et al.*, 2005] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. A hybrid discriminative/generative approach for modeling human activities. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 766–772, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[Lester *et al.*, 2006] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. A practical approach to recognizing physical activities. In KennethP. Fishkin, Bernt Schiele, Paddy Nixon, and Aaron Quigley, editors, *Pervasive Computing*, volume 3968 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg, 2006.