Determining whether Two Devices are on the Same Person using Accelerometers

Midterm Presentation

Zachary Stence and Torry Johnson

Purpose

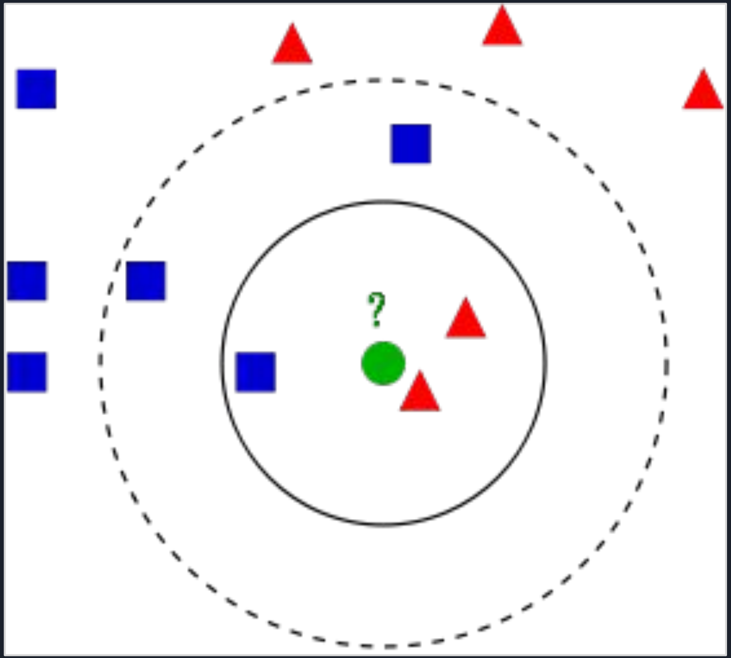
- Wearable IoT devices often have no authentication techniques to verify which user is operating the device.
- Mixing up owners of sensors could cause confusion and health issues to be neglected simply out of oblivion
- Our purpose is to develop a framework of classifying accelerometer streams to later be used to build an authentication system

Challenges

- Imbalanced nature of data
- Time to label data and train/test
- Coherence window has a *huge* impact on accuracy
- Large number of hyperparameters
- Many different models

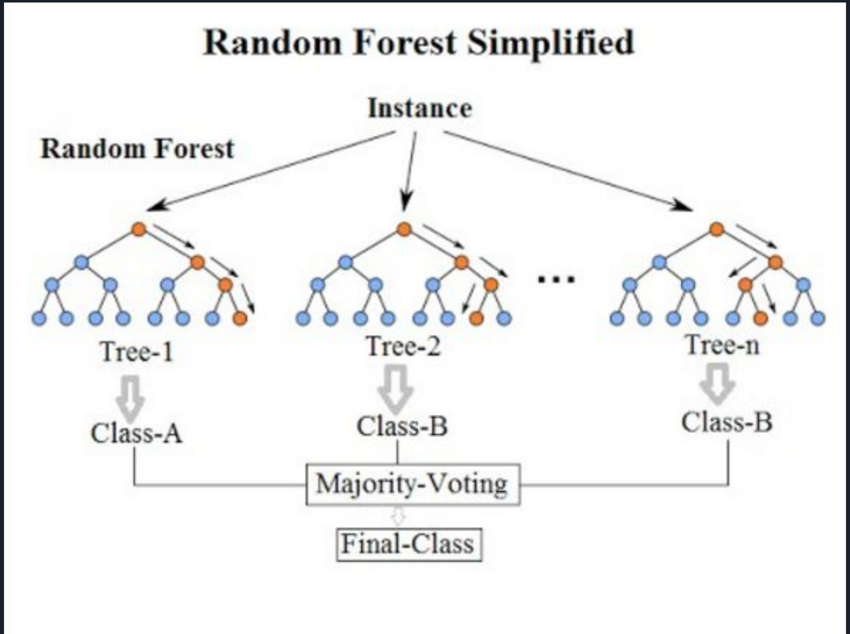
k-Nearest Neighbors

- Train: plot each sample in n-dimensional space
- Test: plot new points and classify based off neighbors
  - Majority vote of k neighbors
  - Determine class



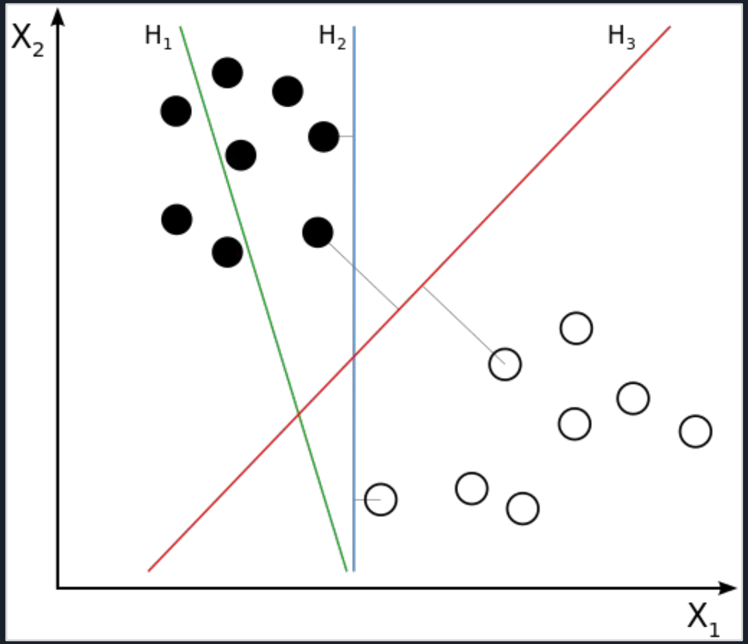
Random Forest

- Create many decision trees
  - Random subsets of data
  - Random subsets of features
- Follow datapoint to leaf
- Take majority vote of trees to classify



Support Vector Machines

- Train: plot each sample in n-dimensional space
- Find hyperplane to separate classes
- Maximize margin to ensure best boundary for classification
- Plot new points and classify



Preliminary Results

- Imbalanced data - always guess false (~96%)
- Short coherence window - no better than guessing (~55%)
- Longer coherence window - better accuracy (~70%)
- Tuned hyperparameters - much better accuracy (~80%)

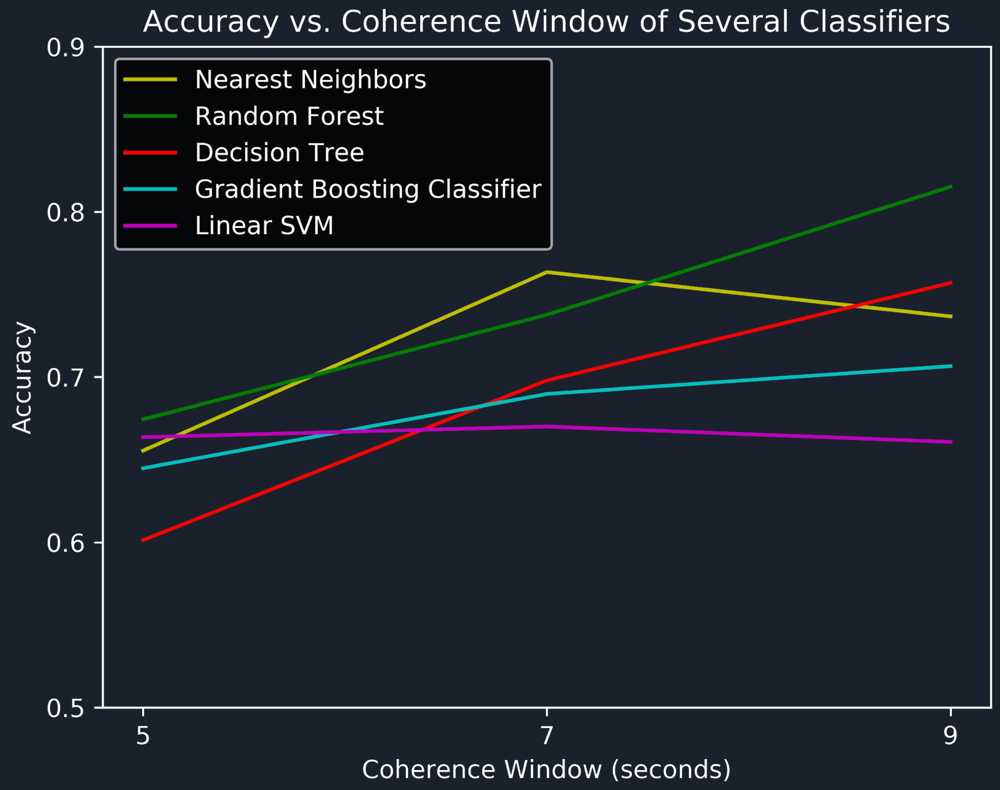
15 people, not balanced

	classifier	train_score	test_score	train_time
3	Nearest Neighbors	0.964450	0.967028	0.234375
4	Random Forest	1.000000	0.967028	63.781250
0	Neural Net	0.963545	0.966889	16.500000
1	Linear SVM	0.963545	0.966889	3.125000
2	Logistic Regression	0.963545	0.966889	0.062500
5	Gradient Boosting Classifier	0.975790	0.962716	16.937500
6	Naive Bayes	0.951892	0.955342	0.343750
7	Decision Tree	1.000000	0.932248	0.390625

15 people, balanced

	classifier	train_score	test_score	train_time
6	Random Forest	1.000000	0.565049	3.265625
5	Nearest Neighbors	0.732134	0.563107	0.000000
4	Logistic Regression	0.570248	0.547573	0.000000
3	Naive Bayes	0.561983	0.541748	0.000000
7	Neural Net	0.560525	0.541748	2.328125
2	Gradient Boosting Classifier	0.996597	0.537864	1.890625
0	Linear SVM	0.561011	0.530097	0.218750
1	Decision Tree	1.000000	0.497087	0.015625

Preliminary Results cont.




```
           predicted
              F     T
actual  F   133     61
        T   52     149
trained Decision Tree in 0.02 s
           predicted
              F     T
actual  F   157     37
        T   89     112
trained Linear SVM in 0.11 s
           predicted
              F     T
actual  F   129     65
        T   32     169
trained Nearest Neighbors in 0.02 s
           predicted
              F     T
actual  F   145     49
        T   54     147
trained Gradient Boosting Classifier in 1.08 s
           predicted
              F     T
actual  F   159     35
        T   40     161
trained Random Forest in 2.44 s
```

Preliminary Results cont.

```
Decision Tree Tuned Hyperparameters
Average accuracy:  0.6910126582278482
Average precision: 0.631180179467885
Average F1 Score:  0.6807788558361927
```

```
Random Forest Tuned Hyperparameters
Average accuracy:  0.8035443037974688
Average precision: 0.7959695725467468
Average F1 Score:  0.8079489235148458
```

```
SVM Tuned Hyperparameters
Average accuracy:  0.6734177215189874
Average precision: 0.7468354430379749
Average F1 Score:  0.6465753424657537
```

```
Nearest Neighbors Tuned Hyperparameters
Average accuracy:  0.8050632911392409
Average precision: 0.7350427350427354
Average F1 Score:  0.8171021377672204
```

Tasks

Tasks	Deliverable	Completion Date		Problems If Missed
		Expected	Actual	
In-depth machine learning research and familiarization	Good understanding of machine learning goals/techniques	5/21	5/21	Trouble understanding breadth/depth of project
Decide how UniMiB SHAR dataset will be used or if we will collect and use our own data	Know which dataset(s) we will use and why	5/23	5/21	Need to decide ASAP
Brush up on Python using online tutorials and coding our own examples	Progress in SoloLearn tutorials	5/23	5/22	Working with data may be difficult
Figure out how to import raw data for use in Python (full_data.mat)	Example program importing a small accelerometer dataset	5/23	5/21	Delay processing, finish ASAP
Determine features of dataset and how data will be used with machine learning algorithms	Outline of data and how it will be used (including specified features)	5/24	5/24	Proposal presentation may lack important info, need to finish ASAP
Project Proposal Submission	Completed Project Proposal	5/25	5/25	Finish ASAP
Determine what the data should look like • how will training/testing take place? • classify data as on same body/not	Conceptual idea of format of data and how it will interact with scikit-learn	5/28	5/28	Proposal presentation may lack important info, need to finish ASAP
Proposal Presentation	Completed Presentation	5/30	5/30	Complete ASAP

Tasks (continued)

Tasks	Deliverable	Completion Date		Problems If Missed
		Expected	Actual	
Get data into correct format (features) ready for use with scikit-learn	Dataset in python in the correct format for scikit-learn	6/8	6/6	Finish before train/test deadline
Outline what our Python code will accomplish	Generic outline of code	6/11	6/8	Difficulty coding program
Code preliminary program in Python for testing one machine learning algorithm	Complete python program	6/14	6/11	First training/testing will be delayed
Train and test our data with at least one algorithm	Results of training/testing (confusion matrix)	6/15	6/11	Finish ASAP. Needed by midterm presentation
Midterm Presentation	Completed presentation	6/20	6/18	Finish ASAP
Midterm Progress Report	Submitted midterm progress report	6/22	6/18	Finish ASAP
Midterm Report	Completed midterm report	6/22		Finish ASAP
Determine which classification algorithm will work the best with our dataset	List of classification algorithms ranked by train-time, test-time, and score	6/25	6/13	Finish soon, may run out of time
Tweaking dataset/algorithms used to solve any unexpected issues that may arise	Fully functional algorithm operating on data	7/7	6/13	Finish ASAP to allow for more in depth analysis
Find best classification algorithm(s)	Confidence level of authentication based on length of data	7/9		Finish ASAP

Tasks (gray cells are additional goals to be achieved if time permits)

Tasks	Deliverable	Completion Date		Problems If Missed
		Expected	Actual	
Obtain statistics about data and algorithm (visualizations) • Confusion matrix • How much time is needed for authentication? • How reliable is authentication? • How do different algorithms perform?	Graphs/plots and information/ statistics about data, algorithms, and overall project	7/11		Finish ASAP for final report
Collect and use our own data	Full dataset of our data collected	N/A		Finish if time permits
Further optimize analysis by refining data and/or algorithms used (hyperparameters)	Machine learning model with improved functionality	N/A		Finish if time permits
Obtain statistics after optimization and observe the increase in performance	Graphs/plots and information/ statistics about data, algorithms, and overall project	N/A		Finish if time permits
Create an example authentication app based off of data collected and analyzed in real-time [7]	Working app prototype	N/A		Finish if time permits
Final Report/Presentation	Completed final report/presentation	7/18		Finish ASAP
Poster Presentation	Completed poster	7/19		Finish ASAP

References

[1] Bianchi, Andrea and Ian Oakley. "Wearable authentication: Trends and opportunities" Usable privacy and security / Alexander De Luca, Emanuel von Zezschwitz. it - Information Technology, 58.5 (2016): 255-262. Retrieved 25 May. 2018, from doi:10.1515/itit-2016-0010

[2] Cornelius, Cory T, and David F Kotz. “Recognizing Whether Sensors Are on the Same Body.” Pervasive and Mobile Computing, vol. 8, no. 6, Dec. 2012, pp. 822–836. ScienceDirect, doi:10.1016/j.pmcj.2012.06.005.

[3] Micucci, Daniela, et al. “UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones.” Applied Sciences, vol. 7, no. 10, 24 Oct. 2017. Applied Sciences, doi:10.3390/app7101101.

[4] Prettenhofer, Peter. “Classification with Scikit-Learn.” DataRobot Blog, 3 Mar. 2014, blog.datarobot.com/classification-with-scikit-learn.

[5] “Scikit-Learn: Machine Learning in Python.” Scikit-Learn, Oct. 2017, scikit-learn.org/stable/index.html.

[6] Taspinar, Ahmet. “Classification with Scikit-Learn.” Ahmet Taspinar, 1 Mar. 2018, ataspinar.com/2017/05/26/classification-with-scikit-learn/.

[7] Xu, Weitao. “Gait-based authentication system on smart wearable devices.” Online video clip. YouTube. YouTube, 22 Aug. 2016. Web. 21 May 2018.