Learning Text with Recurrent Neural Networks in Keras

ZOE KENDALL

Why Deep Learning?

Why Keras?

Instructions for Use

Curabitur volutpat mauris in lorem.

- ivamus turpis
- vulputate at
- sollicitudin id
- ultricies vestibulum
- enim.

Sit amet

consectetuer adipiscing elit. Curabitur sem arcu, tempus ac, gravida suscipit, commodo in, risus. Phasellus eu orci at quam iaculis tempus. Ut at enim. Phasellus diam lectus, pellentesque sed, ullamcorper et, pretium vel, tellus. Donec neque. Maecenas in justo sed arcu aliquet suscipit. Aliquam non est. Quisque pellentesque bibendum mauris. Donec et orci et lectus pharetra posuere. In eleifend, libero vel faucibus vestibulum, neque lectus ultricies dolor, non malesuada leo arcu sit amet erat. Phasellus eros. Vestibulum ornare, lectus et cursus feugiat, risus justo faucibus lacus, sit amet vulputate eros urna vel pede. Vestibulum dapibus dolor eu eros sodales tristique.

Nam Sit

Amet felis in sapien dapibus pharetra. Donec mauris. Suspendisse quis diam at lectus interdum imperdiet. Aliquam cursus metus sed nunc. Sed dignissim tincidunt mi. Vivamus pharetra ultricies quam. Fusce rhoncus. Aliquam blandit molestie tellus. Vivamus sed eros. Praesent accumsan blandit augue. Donec nulla. Aenean pede leo, dignissim id, tincidunt eu, condimentum non, tortor. Vestibulum imperdiet mi quis tortor. Quisque sed elit vitae enim euismod consectetuer. Vivamus interdum, purus in scelerisque aliquam, mi leo cursus nunc, ut vehicula nulla mi nec mauris. Etiam dui. Nulla fermentum ante eget metus molestie pulvinar. Mauris aliquet.

Mauris Hendrerit

Posuere sapien. Maecenas vitae lectus. Proin egestas posuere arcu. Nunc et ipsum sed sapien blandit hendrerit. Vestibulum pulvinar massa vel tortor. Quisque lobortis, odio vitae faucibus feugiat, mi metus pharetra arcu, id ullamcorper ante justo sed lorem. Nam et turpis. Cras interdum rutrum mauris. Fusce est urna, pretium in, vulputate ut, interdum ut, dolor. Duis vitae tellus in mi faucibus porta.

Nam Placerat

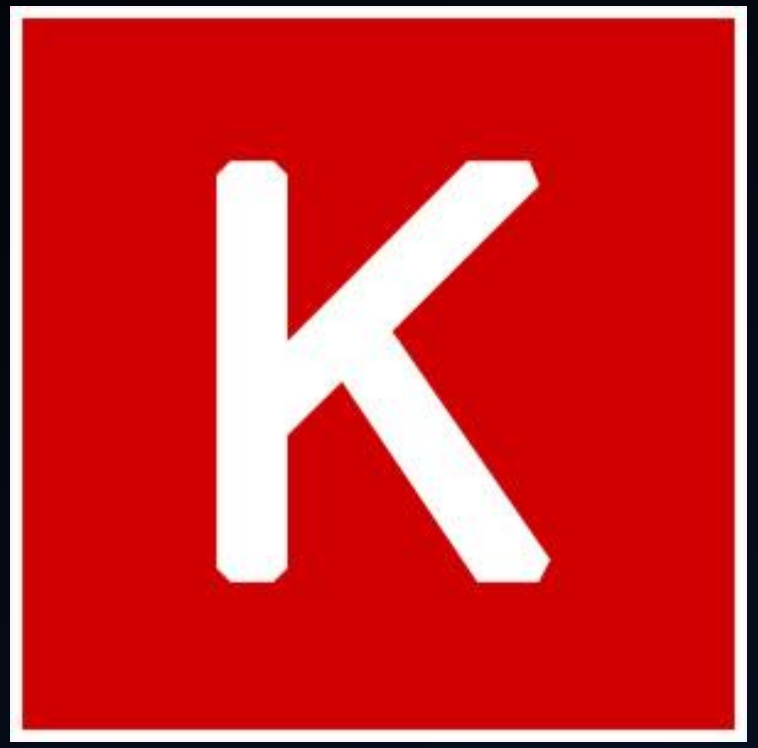
tellus at risus. Phasellus vel metus. Sed a urna. Suspendisse vehicula, arcu vitae dapibus vehicula, justo nulla consectetuer tortor, vitae varius risus purus in turpis. Nunc ut mauris eu diam cursus iaculis. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc at odio dapibus magna lobortis venenatis. Mauris mauris purus, euismod sed, consectetuer eu, tincidunt eget, urna. Maecenas nulla. Proin orci leo, vehicula nec, dignissim rhoncus, luctus nec, odio. Integer pharetra varius nisl. Ut sit amet purus quis est mollis ornare.

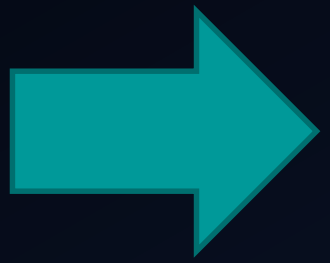
Praesent posuere

Augue vitae pretium venenatis, nunc metus varius velit, a ultricies dolor elit vitae nisl. Proin mollis massa a justo. Aenean orci. Nam elementum fringilla nisl. Nunc eu massa nec eros sodales gravida. Suspendisse ipsum nisl, gravida non, venenatis et, volutpat et, sem. Etiam iaculis nulla. Quisque hendrerit. Cras molestie. Morbi nunc augue, iaculis quis, tincidunt nec, mattis quis, justo. Donec quis nisi a odio blandit tincidunt. Proin non turpis ac ipsum gravida blandit. Integer imperdiet ante quis sapien accumsan dignissim. Nullam nisi tellus, feugiat et, rhoncus blandit, lacinia dapibus, odio.

theano

Why Keras?



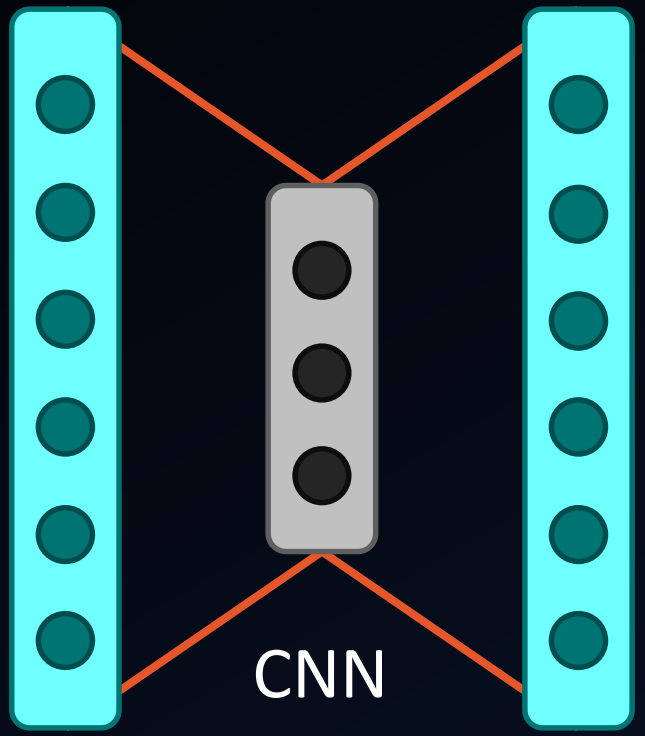


All You Need to Know to do Everything You Want In Four Simple Commands

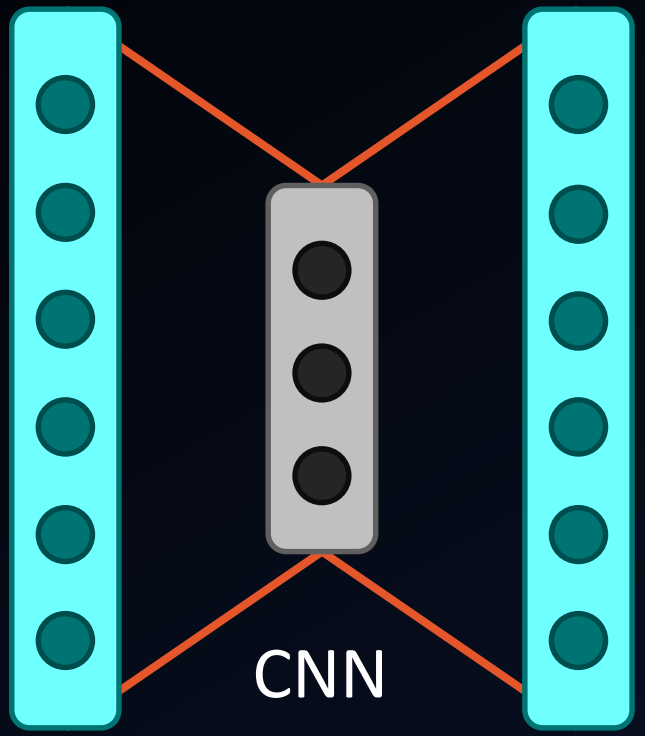
- model = Sequential()
- add()
- fit()
- predict()/evaluate()

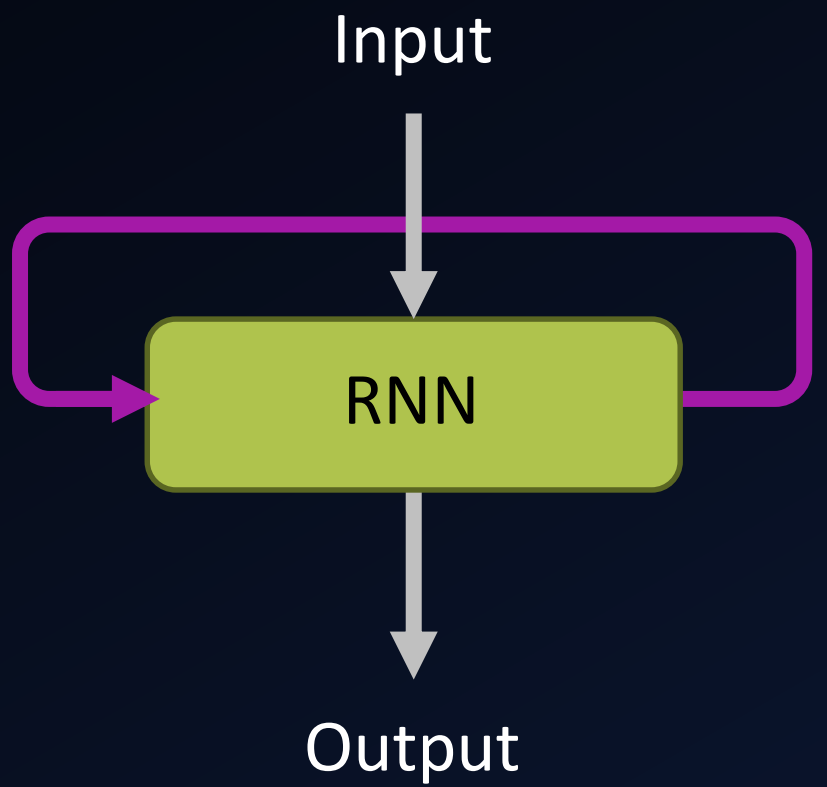
theano

What Can I Build?

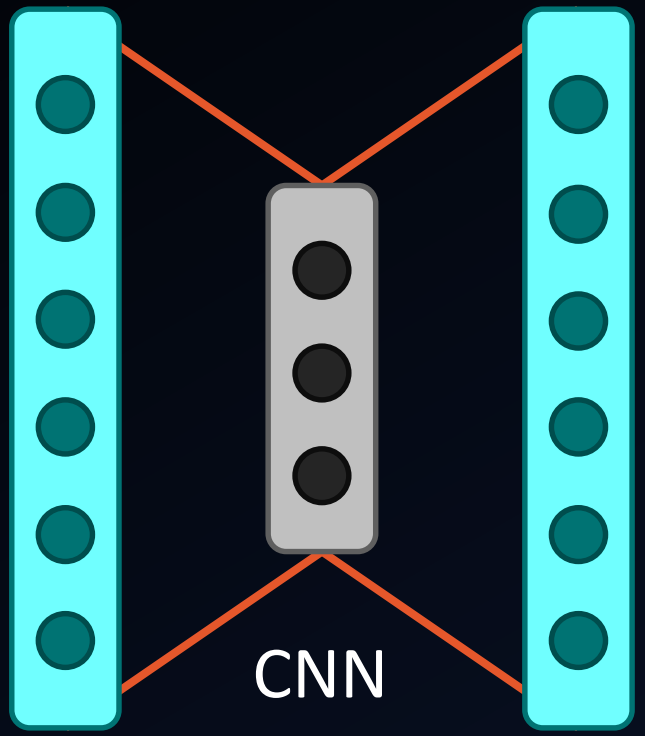


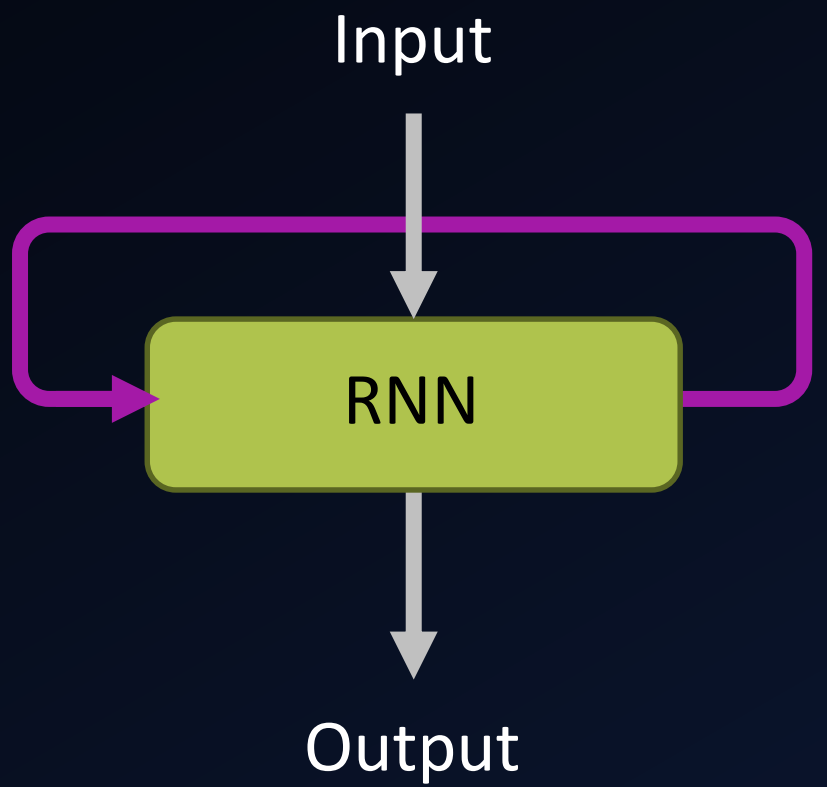
What Can I Build?





What Can I Build?





You can also build custom DNNs with the functional API.

Build a Model as Easy as One, Two, Three

```
37      model = Sequential()
38      # first layer has 12 neurons and expects 8 inputs
39      model.add(Dense(12, input_dim=8, init='uniform', activation='relu'))
40      model.add(Dense(8, init='uniform', activation='relu'))
41      model.add(Dense(1, init='uniform', activation='sigmoid'))
```

Build a Model as Easy as One, Two, Three

```
37    model = Sequential()
38    # first layer has 12 neurons and expects 8 inputs
39    model.add(Dense(12, input_dim=8, init='uniform', activation='relu'))
40    model.add(Dense(8, init='uniform', activation='relu'))
41    model.add(Dense(1, init='uniform', activation='sigmoid'))
```

```
56    model.compile(loss='binary_crossentropy', optimizer='adam',
57                  metrics=['accuracy'])
```

Build a Model as Easy as One, Two, Three

```
37      model = Sequential()
38      # first layer has 12 neurons and expects 8 inputs
39      model.add(Dense(12, input_dim=8, init='uniform', activation='relu'))
40      model.add(Dense(8, init='uniform', activation='relu'))
41      model.add(Dense(1, init='uniform', activation='sigmoid'))
```

```
56      model.compile(loss='binary_crossentropy', optimizer='adam',
57                    metrics=['accuracy'])
```

```
72      model.fit(X, Y, nb_epoch=150, batch_size=10, verbose=1)
```

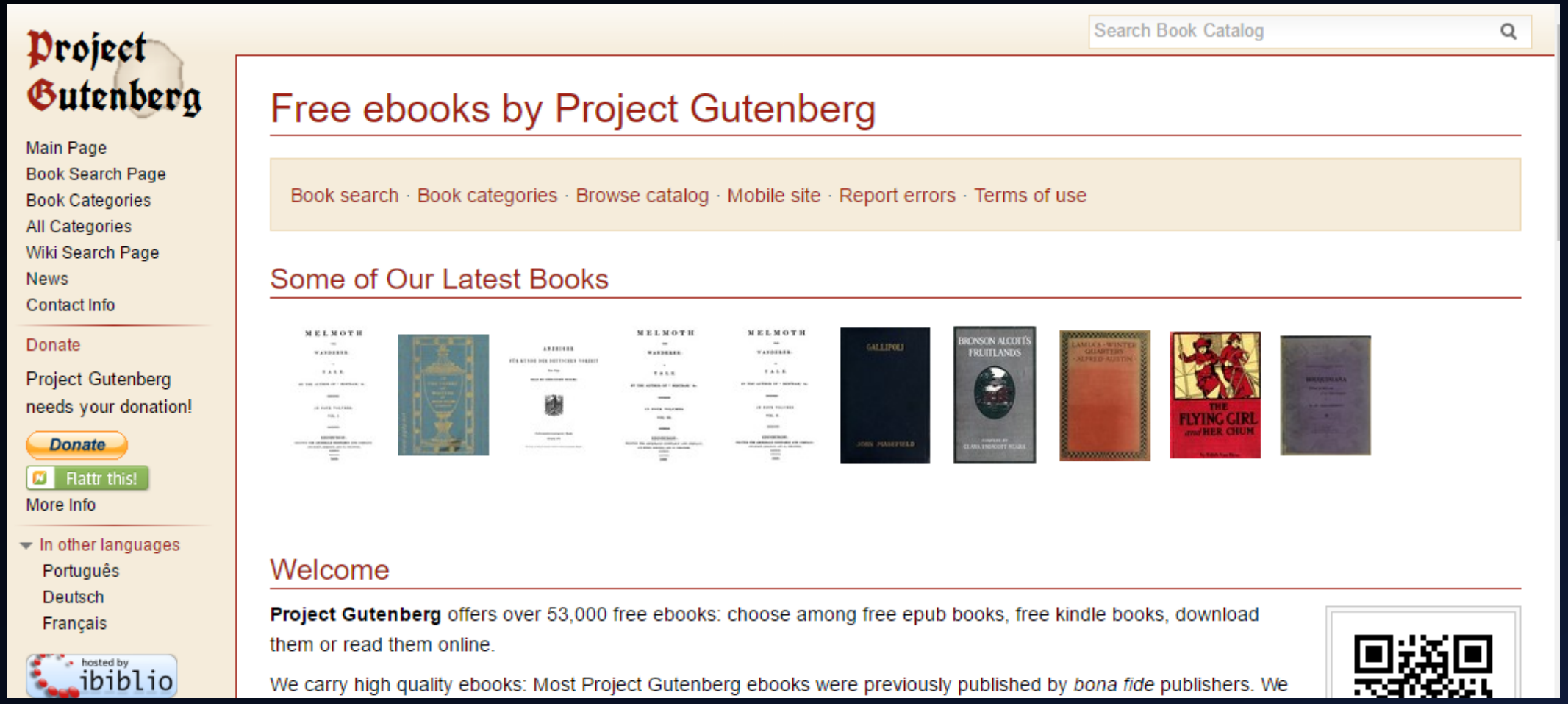
IMDB Sentiment Classification

```
60 def dropLSTM(trainX, trainY, top_words, max_review_length,
61                embed_vec_len=32):
62     model = Sequential()
63     model.add(Embedding(top_words, embed_vec_len, dropout=0.2,
64                         input_length=max_review_length))
65     model.add(Dropout(0.2))
66     model.add(LSTM(100))
67     model.add(Dropout(0.2))
68     model.add(Dense(1, activation='sigmoid'))
69     model.compile(optimizer='adam', loss='binary_crossentropy',
70                   metrics=['accuracy'])
71     print(model.summary())
72     model.fit(trainX, trainY, nb_epoch=3, batch_size=64)
73     return model
```

```
Epoch 1/3
25000/25000 [==============================] - 522s - loss: 0.5732 - acc: 0.6965
Epoch 2/3
25000/25000 [==============================] - 513s - loss: 0.3661 - acc: 0.8436
Epoch 3/3
25000/25000 [==============================] - 520s - loss: 0.3213 - acc: 0.8661
Accuracy: 87.40%
```

Current Plan

- tutorial on building a generative model



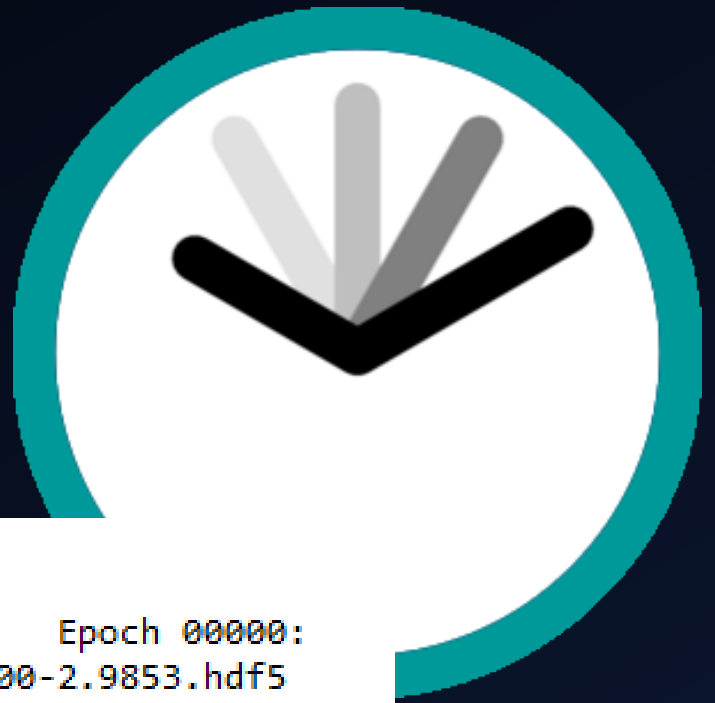
Challenges

Challenges

- g++ vs. Python for Theano

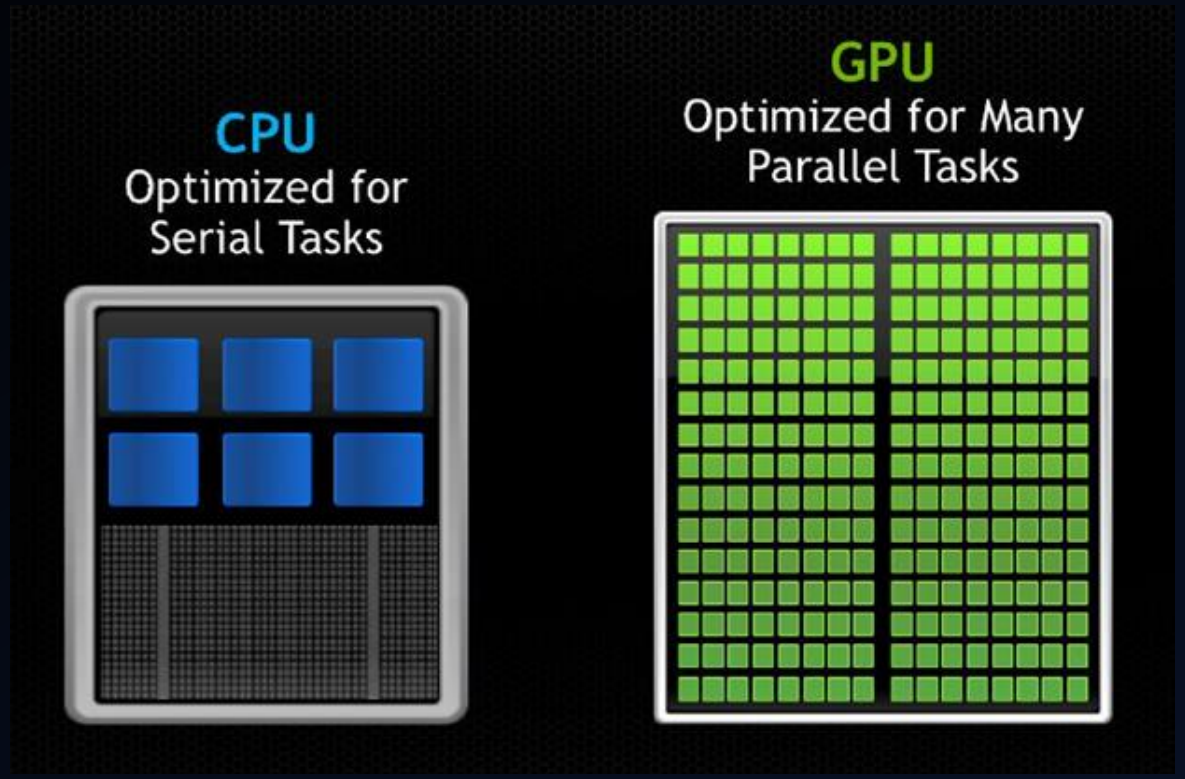
Challenges

- g++ vs. Python for Theano
- computer architecture

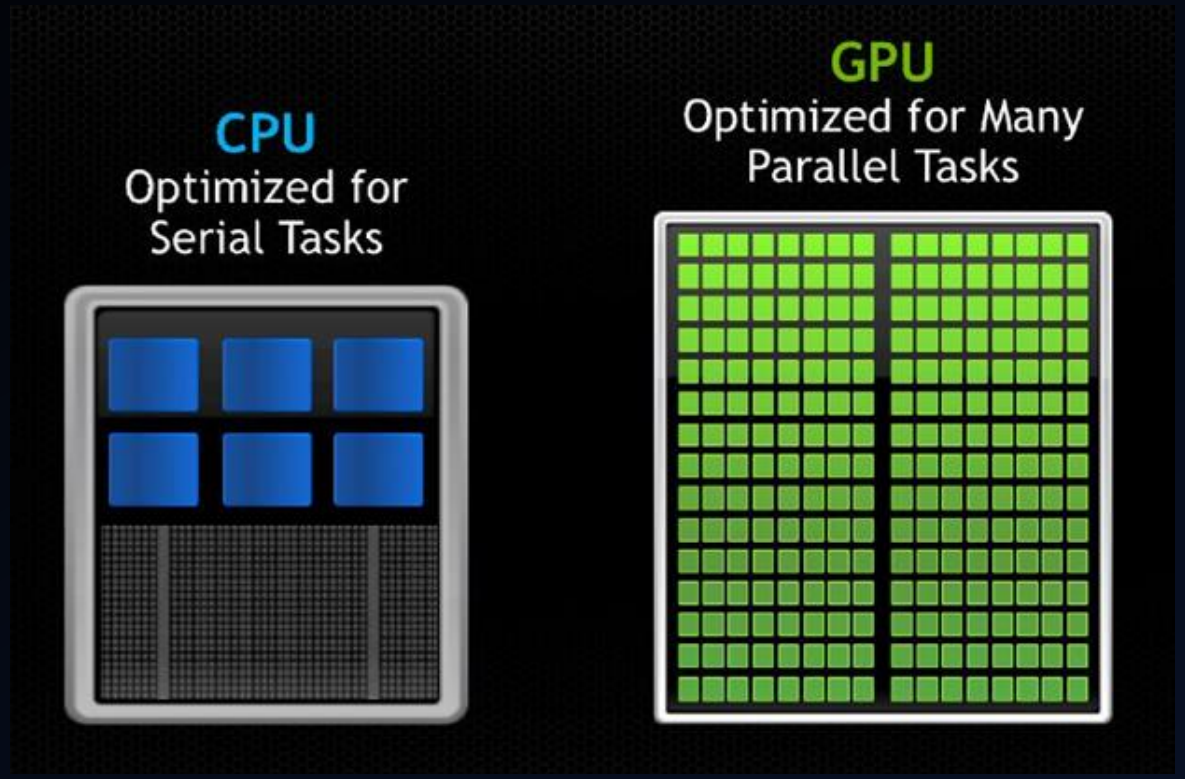


```
Using Theano backend.
Epoch 1/20
144256/144335 [============================>.] - ETA: 0s - loss: 2.9853   Epoch 00000:
loss improved from inf to 2.98533, saving model to weights-improvement-00-2.9853.hdf5
144335/144335 [==============================] - 1673s - loss: 2.9853
Epoch 2/20
144256/144335 [============================>.] - ETA: 0s - loss: 2.7978   Epoch 00001:
loss improved from 2.98533 to 2.79781, saving model to weights-improvement-01-2.7978.hdf5
144335/144335 [==============================] - 1719s - loss: 2.7978
Epoch 3/20
 16256/144335 [==>...........................] - ETA: 1522s - loss: 2.7402
```

Future Work

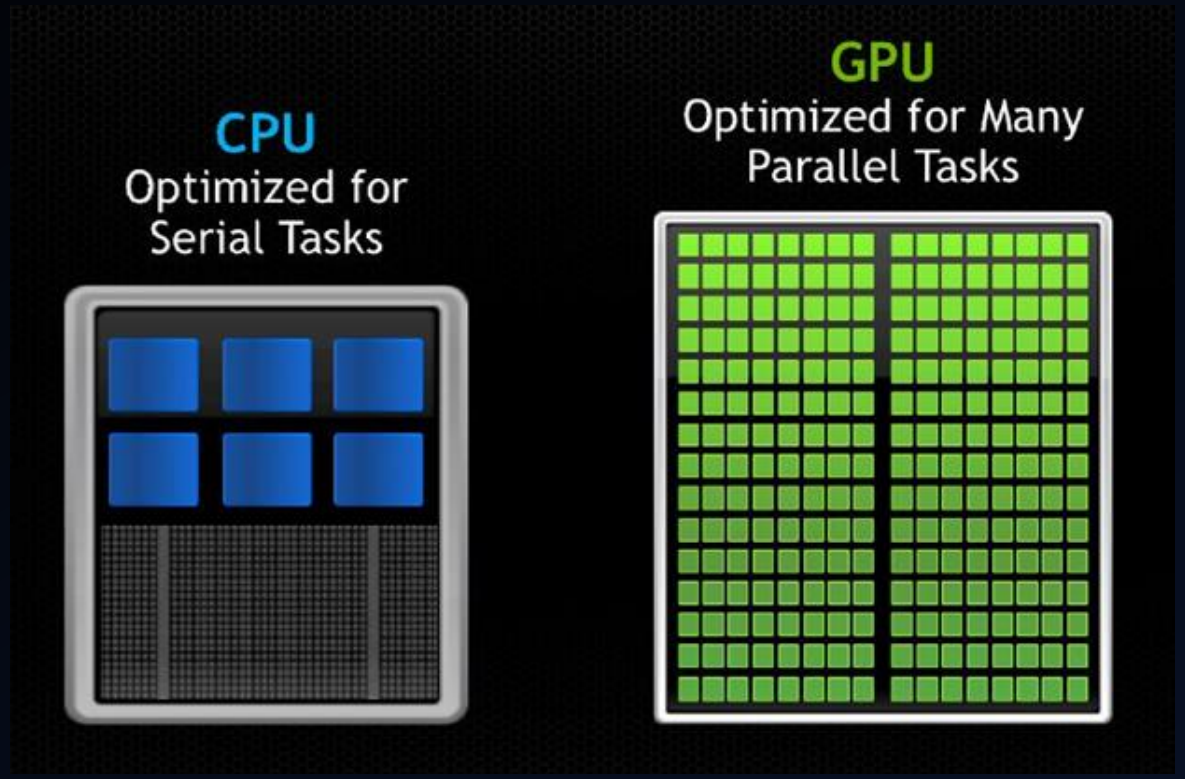

CPU
Optimized for
Serial Tasks
GPU
Optimized for Many
Parallel Tasks

Future Work



theano

Future Work


CPU
Optimized for
Serial Tasks
GPU
Optimized for Many
Parallel Tasks


theano


AWS Compute offerings
EC2
Virtual servers in the cloud
Lambda
Server-less compute platform for Stateless, event-driven code execution
amazon
webservices

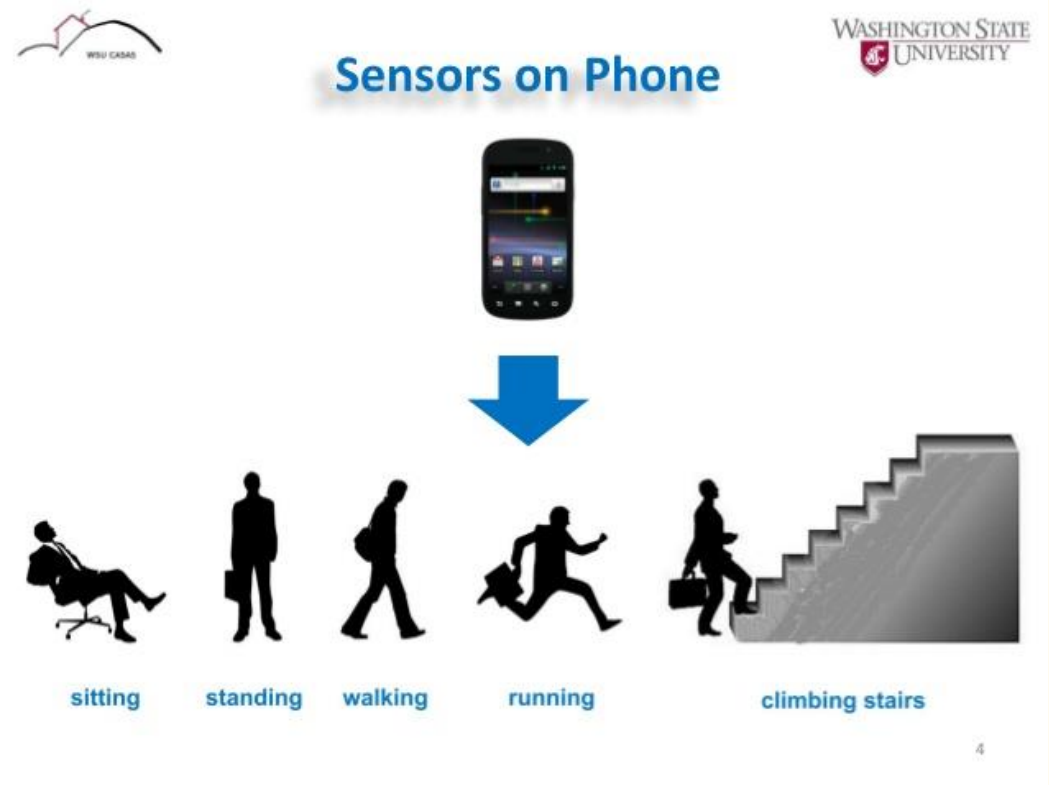
Activity Recognition Using Smartphone Sensors

—

Ravinder Dhesi

Motivation

- Human activity recognition is very useful for things such as elder care and healthcare
- Smartphones becoming able to do this would make it both more accessible and enable more freedom



Ways of analyzing the data

- Supervised learning
- Semi-supervised learning
- Incremental learning approaches
- Unsupervised
- Labels are available but can easily not be used

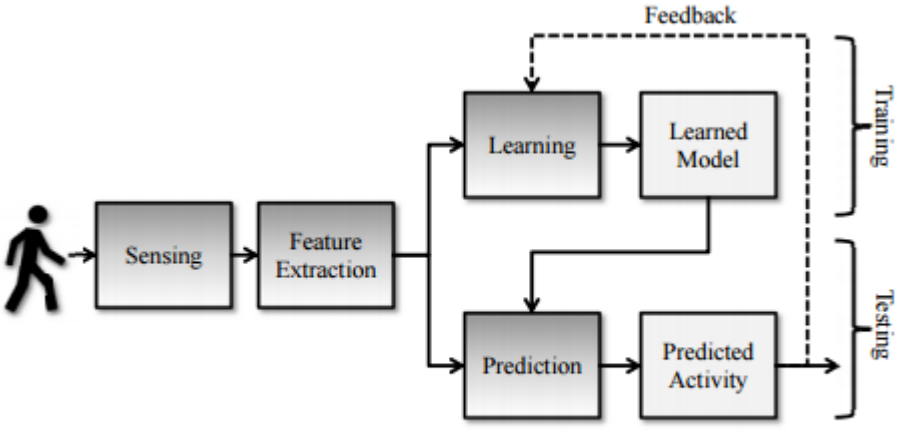


Fig. 1: The Human Activity Recognition Process Pipeline.

Small section of the data

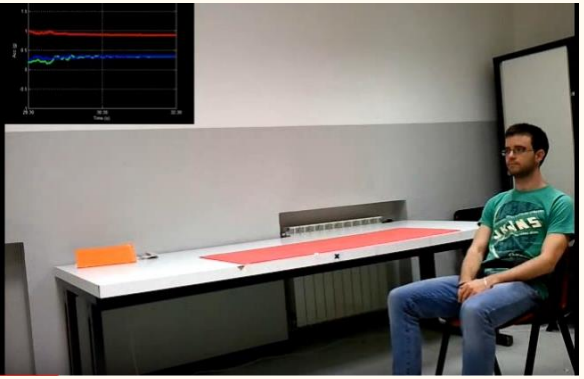
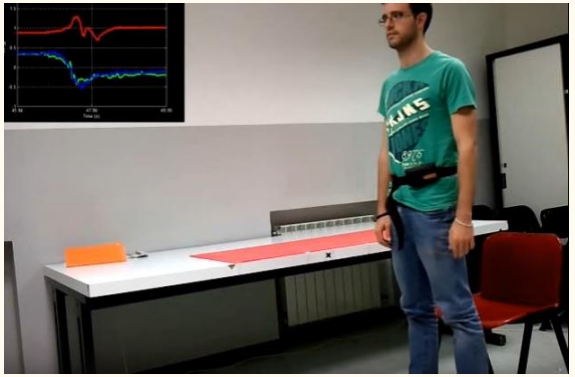
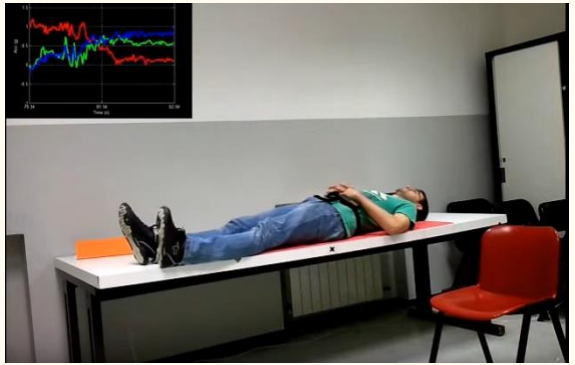
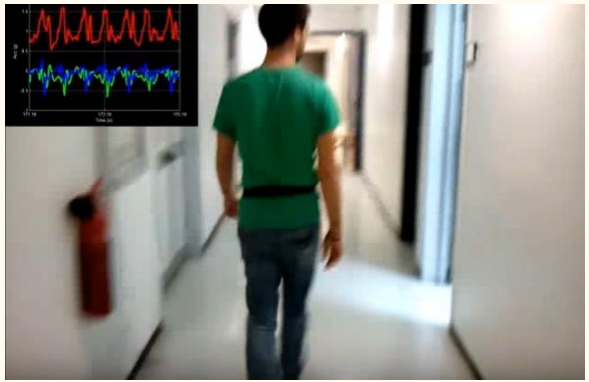
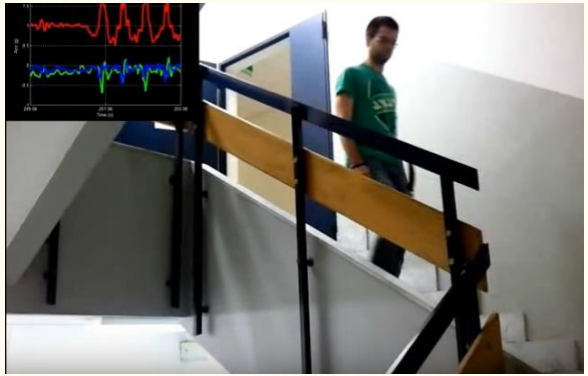
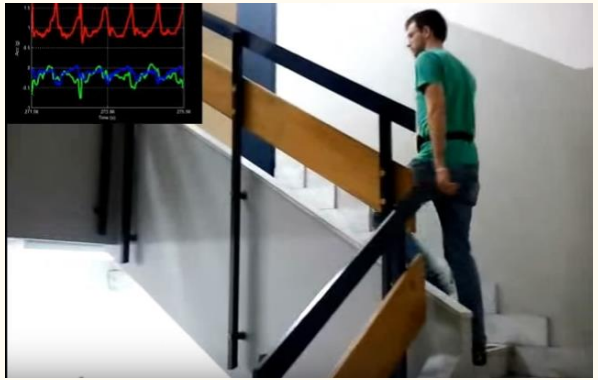
```
 2.8858451e-001 -2.0294171e-002 -1.3290514e-001 -9.9527860e-001 -9.8311061e-001 -9.1352645e-001 -9.9511208e-001
 1.1380614e-001 -5.9042500e-001  5.9114630e-001 -5.9177346e-001  5.9246928e-001 -7.4544878e-001  7.2086167e-001
-9.9236164e-001 -8.6704374e-001 -9.3378602e-001 -7.4756618e-001  8.4730796e-001  9.1489534e-001  8.3084054e-001
2.8225087e-001  9.2726984e-001 -5.7237001e-001  6.9161920e-001  4.6828982e-001 -1.3107697e-001 -8.7159695e-002
9.9348603e-001 -9.9424782e-001 -9.9994898e-001 -9.9454718e-001 -6.1976763e-001  2.9284049e-001 -1.7688920e-001
9.9973783e-001 -9.9973220e-001 -9.9949261e-001 -9.9981364e-001 -9.9968182e-001 -9.9983940e-001 -9.9973823e-001
9.9995513e-001 -9.9991861e-001 -9.9964011e-001 -9.9948330e-001 -9.9996087e-001 -9.9998227e-001 -9.9997072e-001
1.0000000e+000 -1.0000000e+000 -1.0000000e+000 -2.5754888e-001  9.7947109e-002  5.4715105e-001  3.7731121e-001
8.8436120e-002 -4.3647104e-001 -7.9684048e-001 -9.9372565e-001 -9.9375495e-001 -9.9197570e-001 -9.9336472e-001
 2.7841883e-001 -1.6410568e-002 -1.2352019e-001 -9.9824528e-001 -9.7530022e-001 -9.6032199e-001 -9.9880719e-001
-2.1049361e-001 -4.1005552e-001  4.1385634e-001 -4.1756716e-001  4.2132499e-001 -1.9635929e-001  1.2534464e-001
-9.8918458e-001 -8.6490382e-001 -9.5356049e-001 -7.4587000e-001  8.3372106e-001  9.0810964e-001  8.2893499e-001
2.7498054e-002  1.8270272e-001 -1.6745740e-001  2.5325103e-001  1.3233386e-001  2.9385535e-001 -1.8075169e-002
9.9200604e-001 -9.9512320e-001 -9.9996983e-001 -9.9481921e-001 -7.3072160e-001  2.0933413e-001 -1.7811256e-001
9.9954892e-001 -9.9973714e-001 -9.9956575e-001 -9.9990532e-001 -9.9947352e-001 -9.9955418e-001 -9.9960203e-001
9.9996834e-001 -9.9991010e-001 -9.9981369e-001 -9.9992027e-001 -9.9996071e-001 -9.9998672e-001 -9.9995600e-001
1.0000000e+000 -1.0000000e+000 -1.0000000e+000 -4.8167435e-002 -4.0160791e-001 -6.8178329e-002 -4.5855331e-001
4.4149887e-002 -1.2204037e-001 -4.4952188e-001 -9.9033549e-001 -9.9196029e-001 -9.8973198e-001 -9.9448884e-001
 2.7965306e-001 -1.9467156e-002 -1.1346169e-001 -9.9537956e-001 -9.6718701e-001 -9.7894396e-001 -9.9651994e-001
-9.2677626e-001  2.2341317e-003  2.7480687e-002 -5.6728165e-002  8.5533243e-002 -3.2902304e-001  2.7050025e-001
-9.8578618e-001 -8.6490382e-001 -9.5904912e-001 -7.4327710e-001  8.3372106e-001  9.0575280e-001  8.2893499e-001
2.5288704e-001  1.8164885e-001 -1.6930838e-001  1.3200906e-001  8.1973166e-003  1.9332856e-001  7.3717859e-002
9.9765184e-001 -9.9340322e-001 -9.9995494e-001 -9.9398834e-001 -6.6291363e-001  3.2803146e-001 -1.5456001e-001
9.9963593e-001 -9.9967885e-001 -9.9961574e-001 -9.9987958e-001 -9.9910845e-001 -9.9962404e-001 -9.9964327e-001
9.9984140e-001 -9.9992218e-001 -9.9990585e-001 -9.9987361e-001 -9.9999652e-001 -9.9996277e-001 -9.9980377e-001
1.0000000e+000 -8.7096774e-001 -1.0000000e+000 -2.1668507e-001 -1.7264171e-002 -1.1072029e-001  9.0519474e-002
```

Data

1. Inertial sensor data
   - Raw triaxial signals from the accelerometer and gyroscope of all the trials with with participants.
   - The labels of all the performed activities.

2. Records of activity windows. Each one composed of:
   - A 561-feature vector with time and frequency domain variables.
   - Its associated activity label (1-6).
   - An identifier of the subject who carried out the experiment.
   - 7352 labels for train and 2947 for test

Activity Data Collection

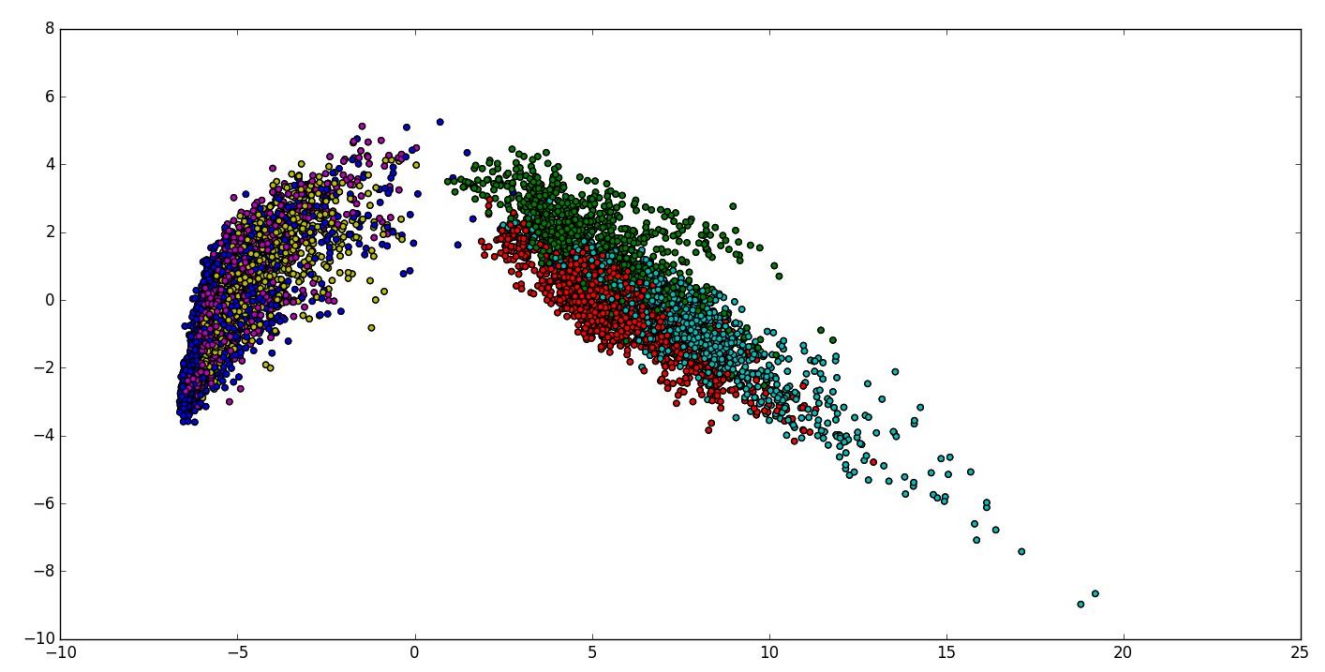







Activities monitored in data set
(https://www.youtube.com/watch?v=XOEN9W05_4A)

Methods

- PCA
- K-means
- 3D PCA
- More methods will probably be applied in the future

Results



PCA

R Walking

G Walking Upstairs

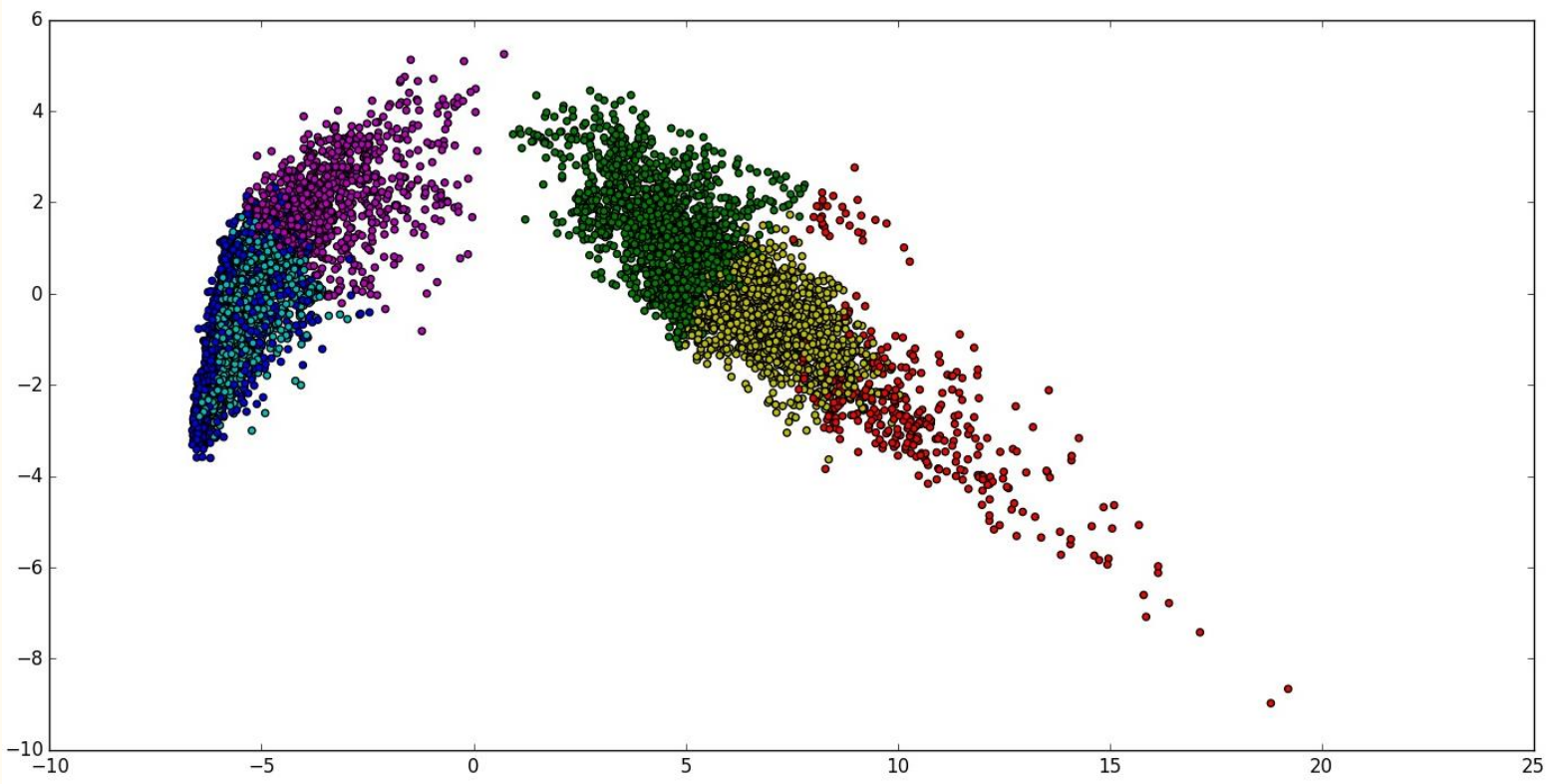
C Walking Downstairs

M Sitting

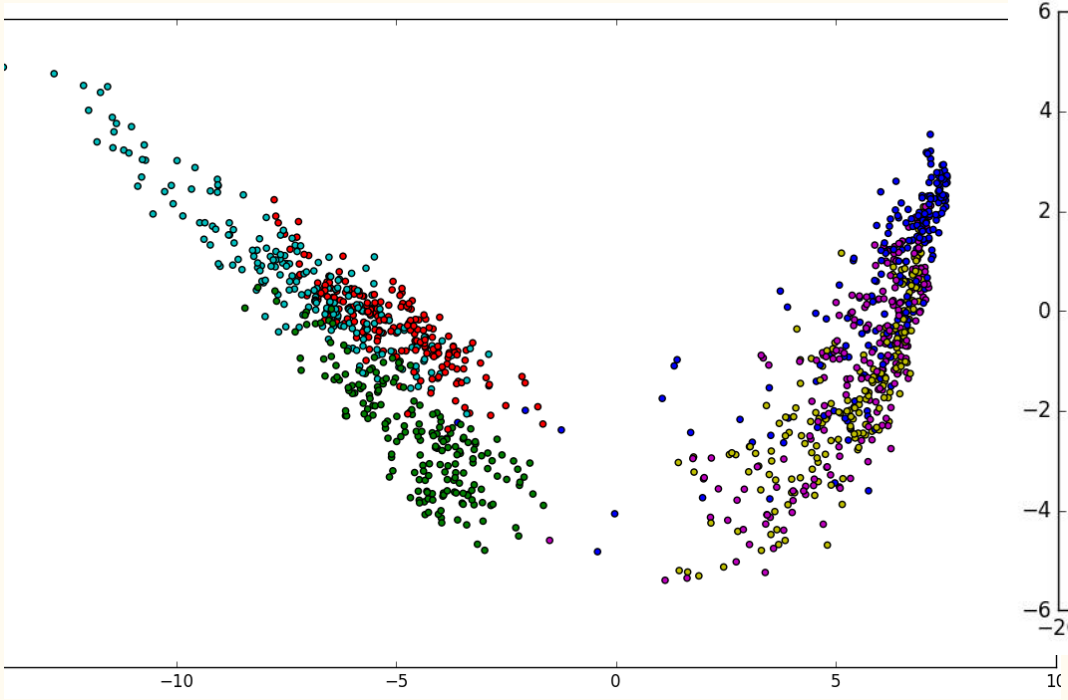
Y Standing

B Laying

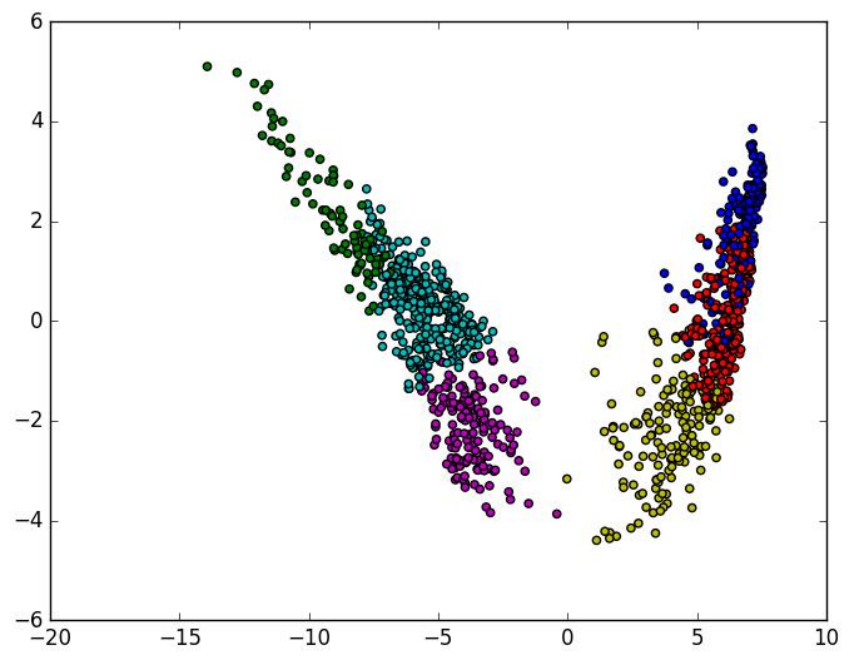
K-Means Attempt



Attempt at reducing data



PCA



K-Means

Current Interpretation and Future Work

- Data looks like it is easier for the sensors to discern movement
- Data overlaps a lot in PCA but uncertain as to whether or not this is due to 2D

Future Work

- Implementing 3D PCA
- Obtain a smaller section of the dataset
- Find more methods to use and compare

References

- Jorge-Luis Reyes-Ortiz, Luca Oneto, Alessandro Ghio, Albert SamÃ¡, Davide Anguita and Xavier Parra. Human Activity Recognition on Smartphones With Awareness of Basic Activities and Postural Transitions. Artificial Neural Networks and Machine Learning â€“ ICANN 2014. Lecture Notes in Computer Science. Springer. 2014.

- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge L. Reyes-Ortiz. Energy Efficient Smartphone-Based Activity Recognition using Fixed-Point Arithmetic. Journal of Universal Computer Science. Special Issue in Ambient Assisted Living: Home Care. Volume 19, Issue 9. May 2013

- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. 4th International Workshop of Ambient Assited Living, IWAAL 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012. Proceedings. Lecture Notes in Computer Science 2012, pp 216-223.

- Jorge Luis Reyes-Ortiz, Alessandro Ghio, Xavier Parra-Llanas, Davide Anguita, Joan Cabestany, Andreu CatalÃ . Human Activity and Motion Disorder Recognition: Towards Smarter Interactive Cognitive Environments. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

Questions? Comments?

Analyzing and Generating Children's Utterances Using Hidden Markov Models

Jessica Tin
CSC 390
December 8, 2016

“...so she grab some food out of her big, blue bag. My blue, big bag. My small little, little, big, little, little bag so I can eat them.”

-Jillian, age 2

Motivation

- PSY/PHI 213 Language Acquisition
- Language follows a set of rules -- how do children learn them?
- Children’s speech does not always follow the rules
- POS tagging
- Hidden Markov Models

- Train a HMM on transcripts of children’s speech to analyze and generate novel utterances

Terminology

- HMM - Hidden Markov Model
- POS tagging - Part-of-Speech Tagging -- labeling words according to their POS (e.g. NN, JJ, DT, etc. (Penn Treebank))
- Token - individual word
- Type - unique word

    e.g. “A rose is a rose is a rose.” = 8 tokens, 3 types

Tools

- CLAN (Computerized Language Analysis)

```
combo @ +u +t*CHI +t%mor +s"adj|*^adj|*" > adj_double.doc
```

- NLTK

```
>>> raw = 'I do not like green eggs and ham, I do not like them Sam I am!'
>>> tokens = word_tokenize(raw)
>>> default_tagger = nltk.DefaultTagger('NN')
>>> default_tagger.tag(tokens)
[('I', 'NN'), ('do', 'NN'), ('not', 'NN'), ('like', 'NN'), ('green', 'NN'),
('eggs', 'NN'), ('and', 'NN'), ('ham', 'NN'), (',', 'NN'), ('I', 'NN'),
('do', 'NN'), ('not', 'NN'), ('like', 'NN'), ('them', 'NN'), ('Sam', 'NN'),
('I', 'NN'), ('am', 'NN'), ('!', 'NN')]
```

- hmmlearn

Data: CHILDES

- Child Language Data Exchange System
- TalkBank, for sharing and studying conversational interactions
- Transcripts of child language
- Some morphologically tagged (POS)

Data: CHILDES

```
2019    *CHI: so she grab some food out of her big, blue bag .
2020    %mor: co|so pro:sub|she v|grab qn|some n|food adv|out prep|of
2021          det:poss|her adj|big cm|cm n|blue n|bag .
2022    %gra: 1|3|COM 2|3|SUBJ 3|0|ROOT 4|5|QUANT 5|3|OBJ 6|3|JCT 7|6|JCT 8|12|MOD
2023          9|12|MOD 10|9|LP 11|12|MOD 12|7|POBJ 13|3|PUNCT
2024    *CHI: bag .
2025    %mor: n|bag .
2026    %gra: 1|0|INCROOT 2|1|PUNCT
2027    *CHI: my blue, big bag .
2028    %mor: det:poss|my n|blue cm|cm adj|big n|bag .
2029    %gra: 1|2|MOD 2|5|MOD 3|2|LP 4|5|MOD 5|0|INCROOT 6|5|PUNCT
2030    *CHI: my small little, little, big, little, little bag so I can eat them .
2031    %mor: det:poss|my adj|small adj|little cm|cm adj|little cm|cm adj|big
2032          cm|cm adj|little cm|cm adj|little n|bag co|so pro:sub|I mod|can
2033          v|eat pro:obj|them .
2034    %gra: 1|12|MOD 2|12|MOD 3|12|MOD 4|3|LP 5|12|MOD 6|5|LP 7|12|MOD 8|7|LP
2035          9|7|ENUM 10|7|LP 11|7|ENUM 12|16|SUBJ 13|16|COM 14|16|SUBJ 15|16|AUX
2036          16|0|ROOT 17|16|OBJ 18|16|PUNCT
```

Method: Pre-processing

- CLAN: strip data

```
combo +s* +o@ -t% +f *.cha > stripped.doc
# strings, header, tiers, file...
```

- NLTK: preprocess stripped data
  - Tokenizing, POS tagging

Method: HMM & Viterbi

- Goal: Choose tag sequence that is most probable given the observed of word sequence
- Probability of a tag depends only on previous tag (bigrams)
- Probability of a word depends only on its tag
- POS Tagging
- Viterbi algorithm: find tag sequence given observations

Method: POS Tagging

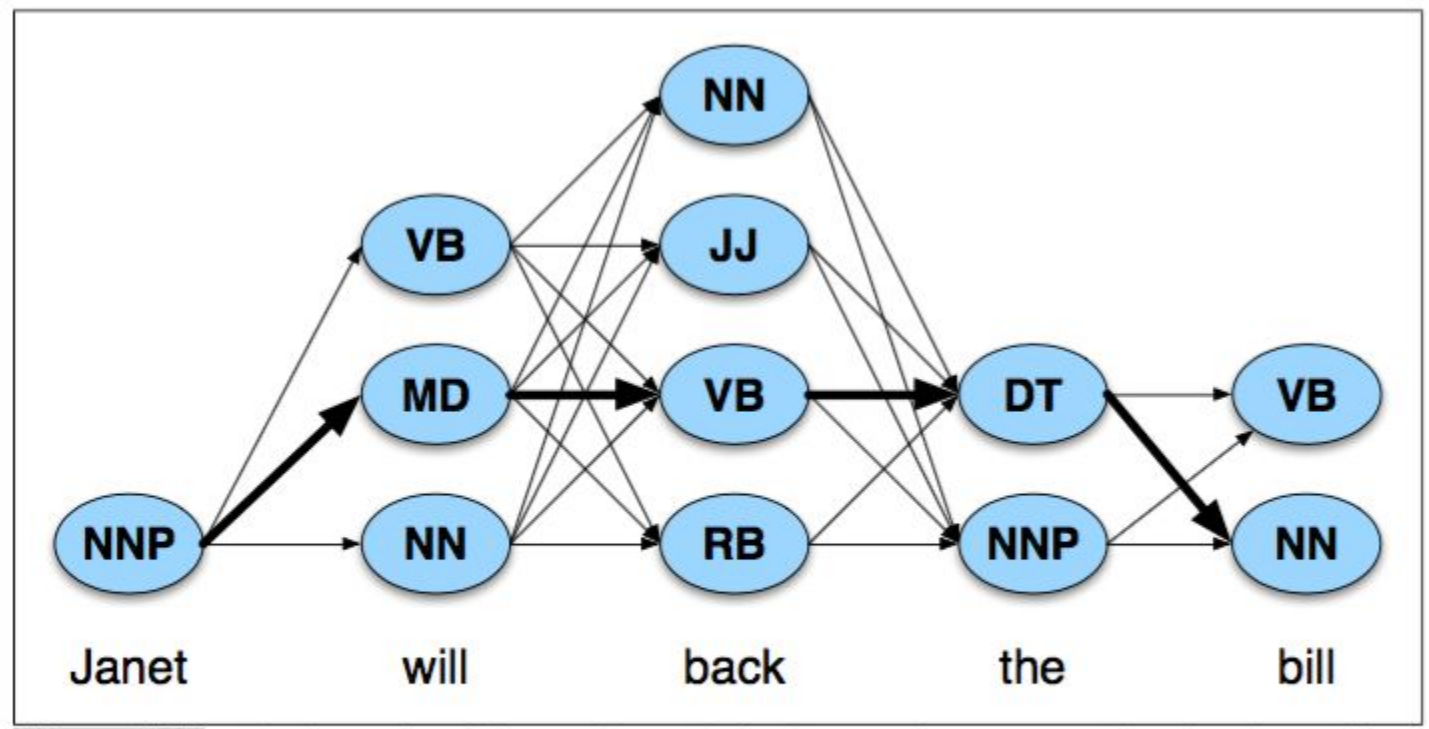


Figure 10.7 A schematic of the tagging task for the sample sentence, showing the ambiguities for each word and the correct tag sequence as the highlighted path through the hidden states.

Method: Viterbi

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*

create a path probability matrix *viterbi*$[N+2,T]$
for each state s **from** 1 **to** N **do** ; initialization step
  viterbi$[s,1] \leftarrow a_{0,s} * b_s(o_1)$
  backpointer$[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ; recursion step
  for each state s **from** 1 **to** N **do**
    $viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
    $backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s}$
$viterbi[q_F,T] \leftarrow \max_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$; termination step
$backpointer[q_F,T] \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T] * a_{s,q_F}$; termination step
return the backtrace path by following backpointers to states back in time from *backpointer*$[q_F,T]$

Figure 10.8 Viterbi algorithm for finding optimal sequence of tags. Given an observation sequence and an HMM $\lambda = (A,B)$, the algorithm returns the state path through the HMM that assigns maximum likelihood to the observation sequence. Note that states 0 and q_F are non-emitting.

Interpretation and Results

- Hidden states: POS tags
- Observations (emissions): Words
- HMM: probability of sequences of tags
- NLTK analysis: most common types

To do: produce probability matrices, use to generate sentences

Future work: compare to adult speech?

Questions?

Thank you!

Using Machine Learning to Discover People:

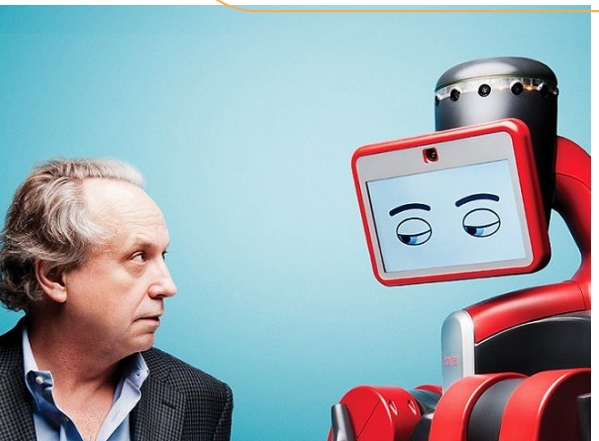
A Discussion About K-means and Hierarchical Clustering Algorithms

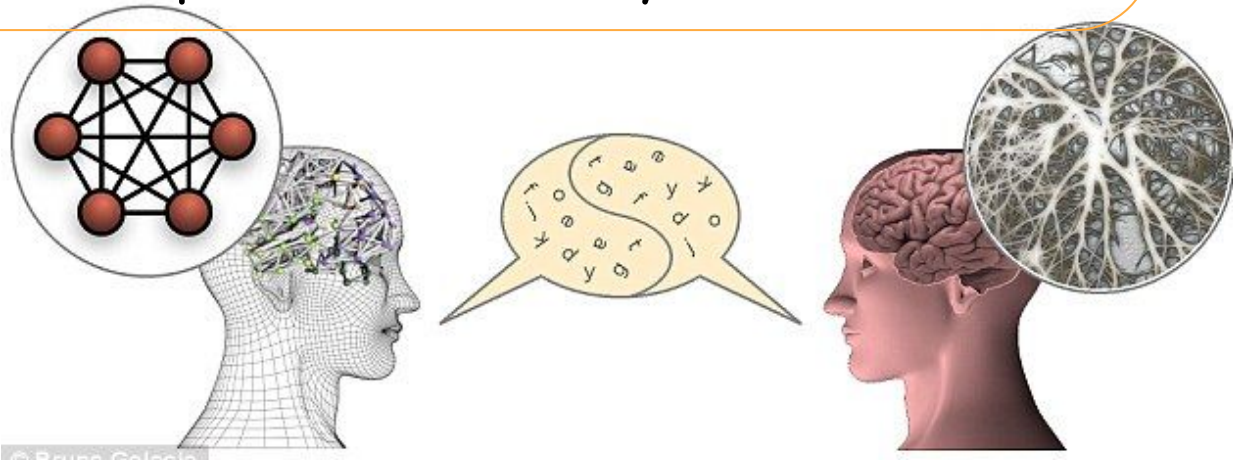
Maria Xu

Final Project

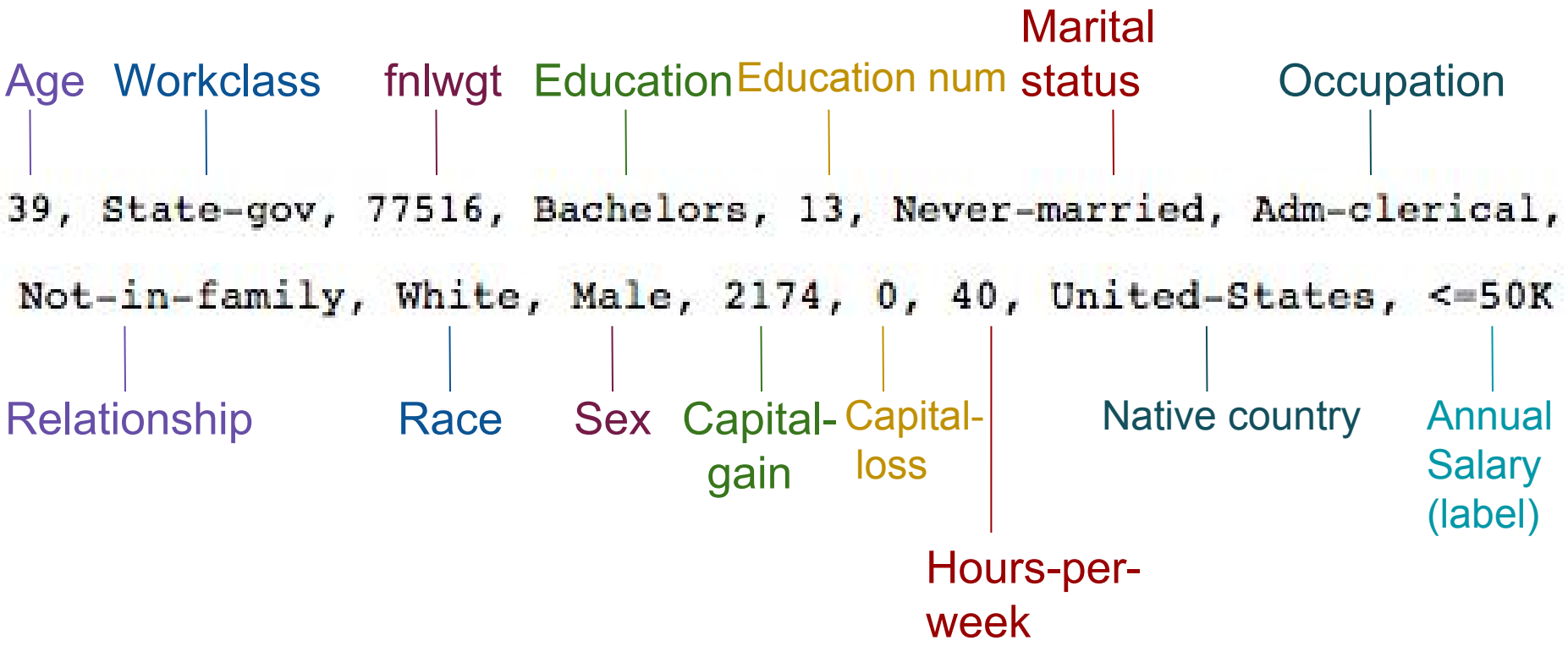
Motivation

- How would algorithms analyze human?
- How would different machine learning algorithms read and interpret the data?
- How do different human factors (e.x. Sex, gender, race, marital-status, etc) relate to the annual salary they earn? Could we predict the salary?





Data


Age
Workclass
fnlwgt
Education
Education num
Marital status
Occupation
39, State-gov, 77516, Bachelors, 13, Never-married, Adm-clerical,
Not-in-family, White, Male, 2174, 0, 40, United-States, <=50K
Relationship
Race
Sex
Capital-gain
Capital-loss
Hours-per-week
Native country
Annual Salary (label)

age: continuous.
workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt: continuous.
education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num: continuous.
marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex: Female, Male.
capital-gain: continuous.
capital-loss: continuous.
hours-per-week: continuous.
native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
>50K, <=50K.

Method

First step: clean data

```
clean_data = []

for line in data_file:
    # remove data that contains "?"
    if "?" not in line:
        data_list = line.split(', ')
        clean_data.append(data_list)
```

After cleaning: the data size is 30162

Method

Second step: pre-process the data

- Replace categorical data with numerical labels
- Save all numerical variables and categorical data separately in two lists for later use.

Third step: run UPGMA on the entire dataset

- 1) create 30162 x 30162 distance matrix
- 2) iteration: merge the two “clusters” that have the shortest distance until there’s only 1 cluster left.
- 3) start printing out the groupings when the cluster numbers <= 5 (this number could be tested).
- 4) analyze the output

Method

Fourth step: run k-means and UPGMA separately only on the numerical variables

- **Goal**: compare k-means and UPGMA algorithms output
- K-means: use “elbow” plot to determine the optimal “k”
- UPGMA: when in “k” clusterings, compare the output with k-means clustering.

Fifth step: run k-means on the numerical variables and UPGMA on categorical variables and compare the outputs with step 2 (running UPGMA on whole).

- **Goal**: to see if the method of running two clustering algorithms on different parts of one dataset would work out for interpreting the dataset.

Results and Other Options

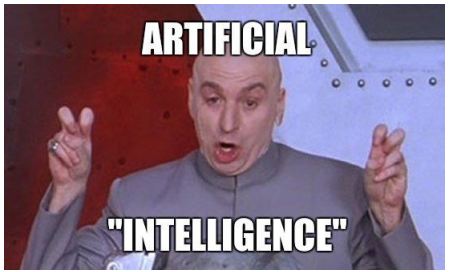
Part1: analyze UPGMA output for the entire dataset.

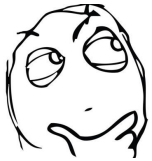
Part2: analyze k-means and UPGMA outputs on the numerical variables, and compare and discuss the pros and cons of two algorithms.

Part3: analyze the k-means output on the numerical variables and UPGMA output on categorical variables, and see how they would help to get some better insights about the dataset.

Part4: discuss the limit of algorithms on analyzing people.

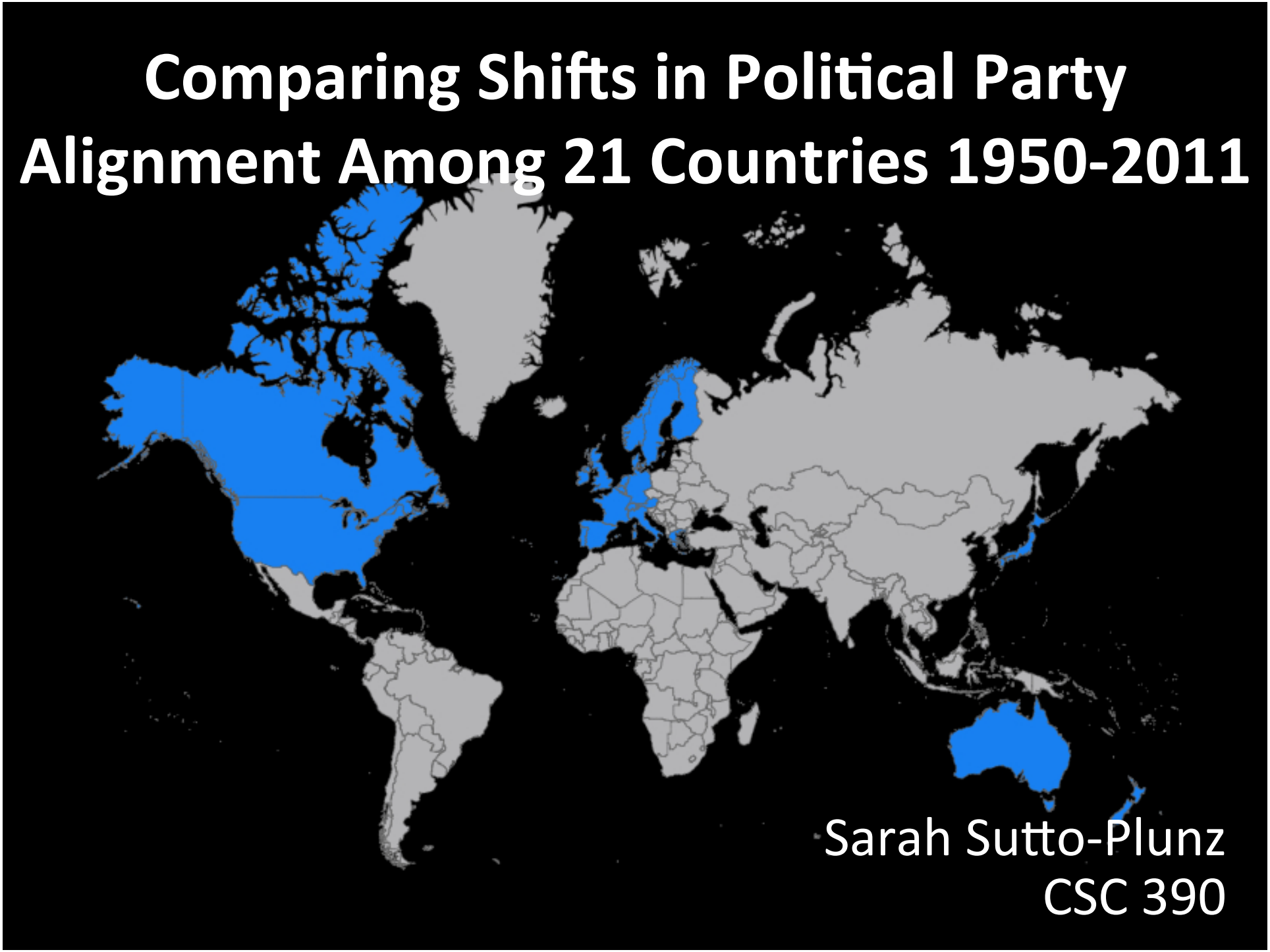
Part 5 (optional): predict the annual salary.





QUESTIONS?

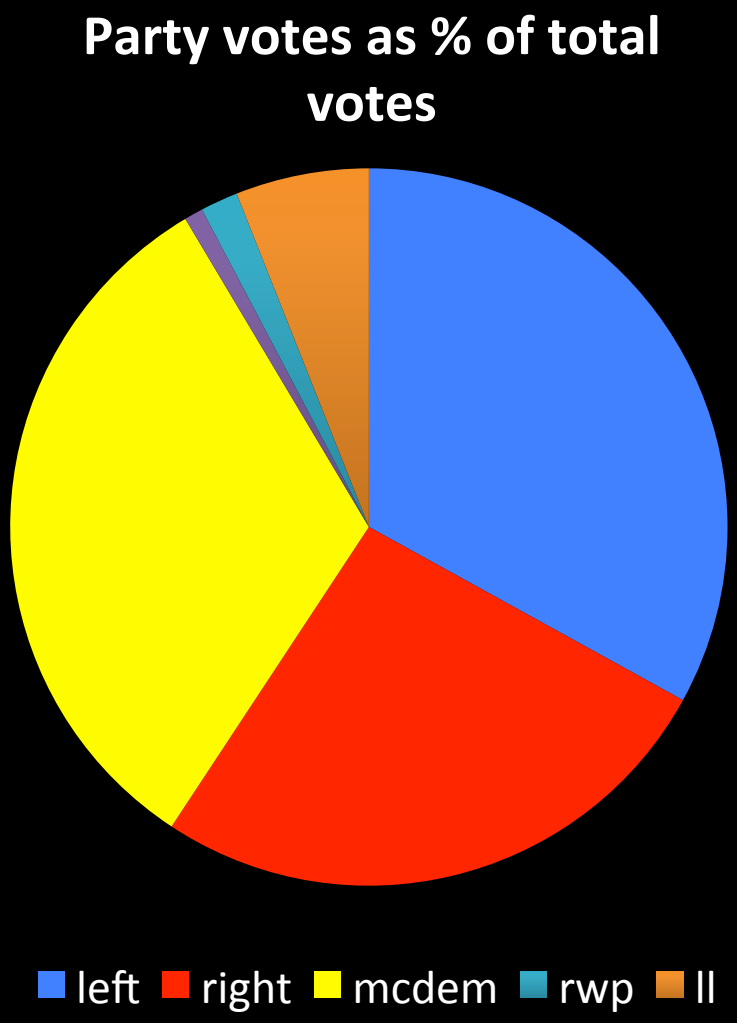
Comparing Shifts in Political Party Alignment Among 21 Countries 1950-2011



Sarah Sutto-Plunz
CSC 390

Motivation

- Straight-forward motivation:
  - To observe the differences between the political makeup of all the countries, to potentially discover trends across the years
- Why does this matter?
  - Emergence or growth of extremist groups or parties
  - Historic events and their quantifiable impact
- Key terminology
  - Defined with the data



Data

- Dataset:
  - Comparative Political Parties Dataset compiled by Duane Swank
- Classifications of political party:
  - Based off of Castles and Mair (1984)
- Number of data points (m) – 1302
- Number of features (p) – 29
- Labels – COLID, ELECTY, ELMON, ELDAY
- All of the feature values are percentages, making them easily comparable

Data

1	1950	0	0	0	0	0	39	47	100	45	60	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1951	1	4	28	0	0	43	48	100	51	57	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1952	0	0	0	0	0	45	49	100	55	55	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1953	0	0	0	0	0	45	49	100	55	55	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1954	1	5	29	0	0	47	50	100	53	53	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1955	1	12	10	0	0	47	50	100	53	53	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1956	0	0	0	0	0	39	46	100	61	61	53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1957	0	0	0	0	0	39	46	100	61	61	53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1958	1	11	22	0	0	39	46	100	61	61	53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1959	0	0	0	0	0	38	43	100	62	62	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1960	0	0	0	0	0	38	43	100	62	62	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1961	1	12	9	0	0	39	43	100	61	61	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1962	0	0	0	0	0	50	48	100	50	50	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1963	1	11	30	0	0	49	48	100	50	50	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1964	0	0	0	0	0	42	46	100	58	58	53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

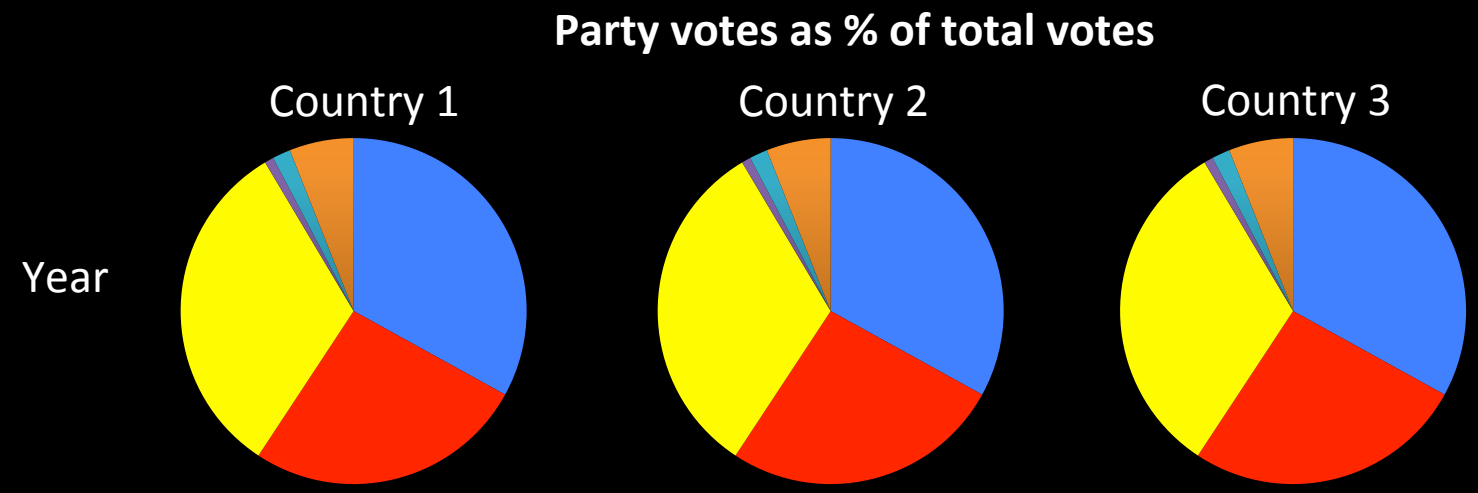
- **C** (party cabinet portfolios as % of all cab port), **GS** (governing party seats as a % of all legislative seats), **S** (party legislative seats as % of all leg. seats), **V** (party votes as % of total votes)
- Other: **LEFT**, **RIGHT**, **TCDEM** (total Christian democratic party), **MCDEM** (centrist Christian democratic party), **CENT** (center party), **RWP** (right-wing populist governing party), **LL** (left libertarian, "new-left" parties)

Methods

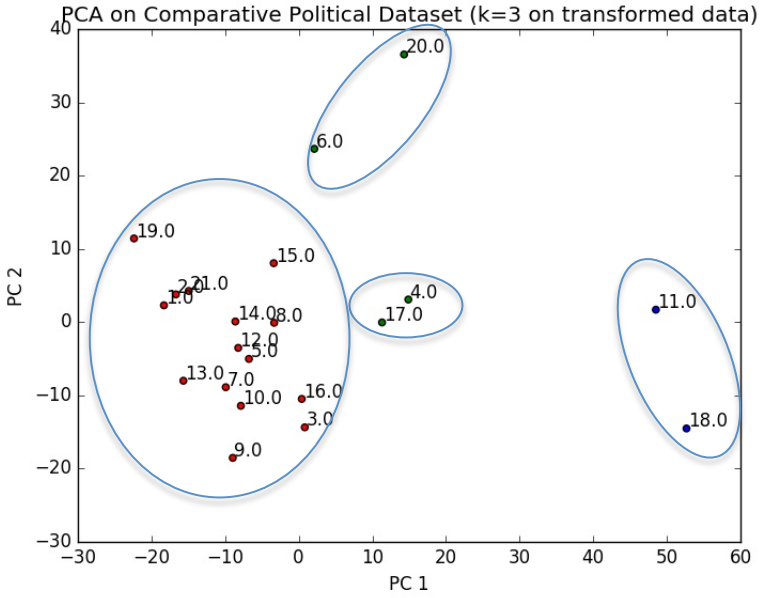
- Plan on starting with k-means, and PCA, largely following HW4

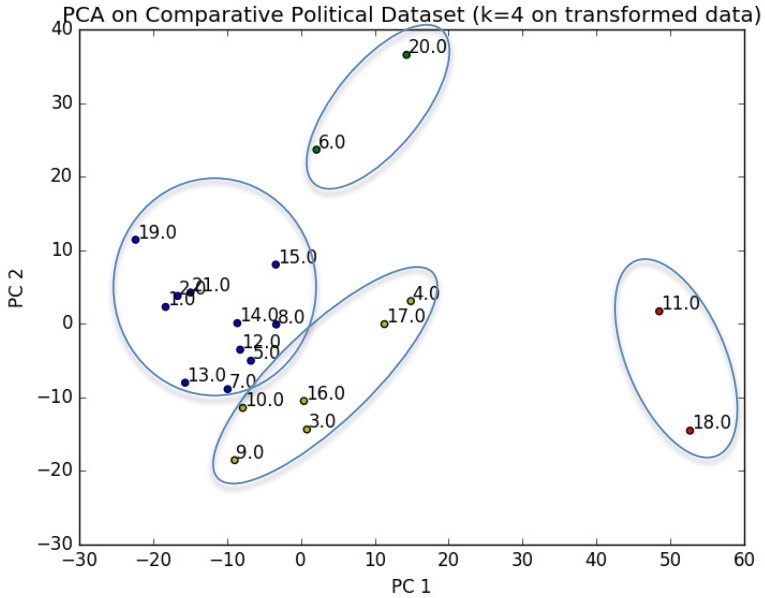
$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

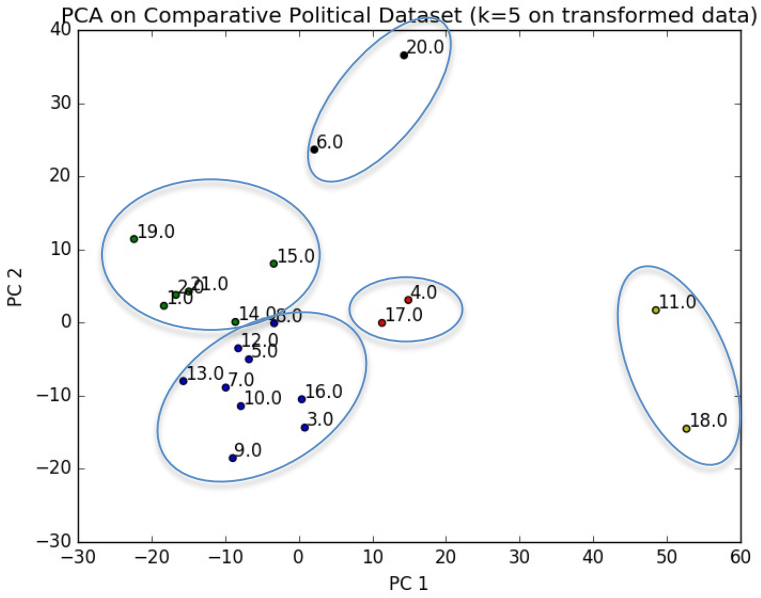
- If I have time, I would also be interested in seeing if something like a time series would work, so something like an HMM
- Essential that percentages are used in the dataset
- Distance between data points

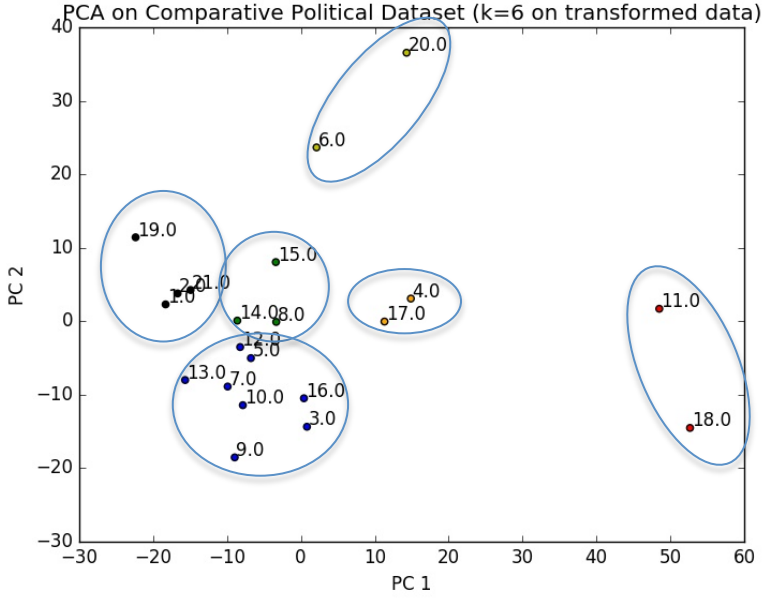


Results


PCA on Comparative Political Dataset (k=3 on transformed data)
PC 1
PC 2
20.0
6.0
19.0
15.0
21.0
4.0
17.0
14.0
8.0
12.0
5.0
13.0
7.0
10.0
16.0
3.0
9.0
11.0
18.0


PCA on Comparative Political Dataset (k=4 on transformed data)
PC 1
PC 2
20.0
6.0
19.0
15.0
21.0
4.0
17.0
14.0
8.0
12.0
5.0
13.0
7.0
10.0
16.0
3.0
9.0
11.0
18.0


PCA on Comparative Political Dataset (k=5 on transformed data)
PC 1
PC 2
20.0
6.0
19.0
15.0
21.0
4.0
17.0
14.0
8.0
12.0
5.0
13.0
7.0
10.0
16.0
3.0
9.0
11.0
18.0


PCA on Comparative Political Dataset (k=6 on transformed data)
PC 1
PC 2
20.0
6.0
19.0
15.0
21.0
4.0
17.0
14.0
8.0
12.0
5.0
13.0
7.0
10.0
16.0
3.0
9.0
11.0
18.0

Supervised v Unsupervised

- For this project, the classification itself will be unsupervised
- The labels (both the country ID and the information to do with election year) will be used to analyze the data and for the visualization as seen on the previous slide

Interpretation and Future Work

- What do the clusters mean?
  - The clusters are formed from the percent of party distribution within them for a specific year
  - Not exactly sure how to make sense of it all, hard to know until the entire time series can be seen
- Future Work
  - Not just k-means, this is a starting point
    - UPGMA, HMM
  - Could become predictive measure
  - Could lend more depth to the current historical narrative
- Questions or Comments?

Sources

- https://www.amcharts.com/visited_countries/#AT,BE,CH,DE,DK,ES,FI,FR,GB,GR,IE,IT,NL,NO,PT,SE,CA,US,JP,AU,NZ
- http://www.nsd.uib.no/macrodataguide/set.html?id=5&sub=1
- http://www.marquette.edu/polisci/documents/Party19502011code.pdf
- Duane Swank, (2013). Comparative Political Parties Dataset: Electoral, Legislative, and Government Strength of Political Parties by Ideological Group in 21 Capitalist Democracies, 1950-2011. Electronic Database, Department of Political Science, Marquette University, http://www.marquette.edu/polisci/faculty_swank.shtml).
- Castles, Frances G., and Peter Mair. 1984. “Left-right political scales: some ‘expert’ judgements”. European Journal of Political Research 12 (1): 147-157.