Author Age and Gender Prediction from Written Samples

Matthew Bowers, Jayam Patel, Jonas Rogers, Qing Xu

Agenda

- Motivation / Goals
- Background / Related Work
- Methodology
  - Feature Extraction and Selection
  - Classifiers
- Results
- Conclusions

Motivation

- Profiling anonymous writers can help legitimize or delegitimize the writing
- Allow social networks to restrict users who do not fall in the desired age group
- Allow marketing companies to know about the people who like or dislike their product based on the reviews

Goals

- Detect age of author of writing sample within decade (10s, 20s, 30s)
- Detect gender of author
- Determine best classifiers by comparing performance
- Compare word based with character based feature extraction

Background

Feature Extraction

- Content based
- POS tagging
- Sentence length average
- Vocabulary diversity
- Stop words
- N-grams

Classifiers

- Naive Bayes
- Decision Tree
- Linear Regression
- k - Nearest Neighbors
- Support Vector Machine
- Maximum Entropy

Data Source

- Blog Posts
- Tweets
- Phone Conversations
- Novels
- Wikipedia Articles
- Product Reviews

Methodology

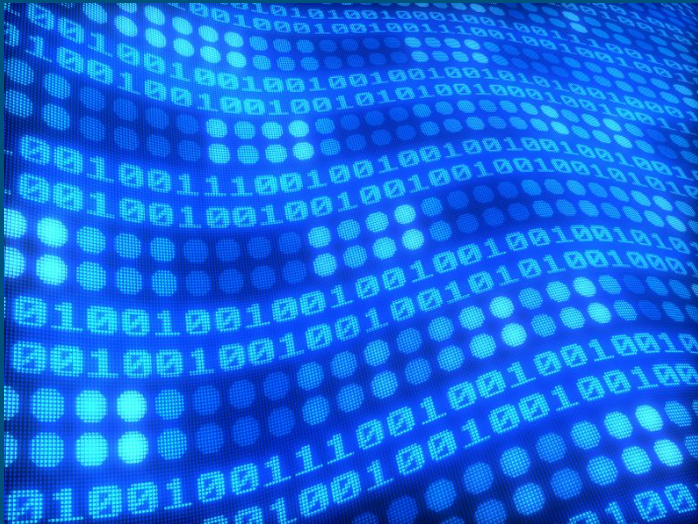
Data

Chatting records from PAN 2013 Author Profiling

Used Test Corpus 2:

- Used English language text
- ~25,000 authors
- Dozens of sentences per author
- Categorized by age and gender

We split this data into a training set and a test set



Feature Extraction

Used sklearn to implement extraction strategies

Extracted N-grams

Words

- Mono-grams
- Bi-grams

Characters

- Tri-grams
- Quad-grams

Feature Selection

Filtered out stop words

Used Variance Threshold method which eliminates features outside of a set variance

Variance Threshold = 0.16

Word N-grams had 2322 features after selection

Character N-grams had 20566 features after selection

Tf-idf

Transforming the data to this form involves multiplying the term frequency with the inverse document frequency.

$$idf(t) = log\frac{n_d}{1 + df(d,t))}$$

Rows are normalized to have unit norm

$$v_{norm} = \frac{v}{||v||^2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + ... + v_n^2}}$$

Classifiers — Decision Tree

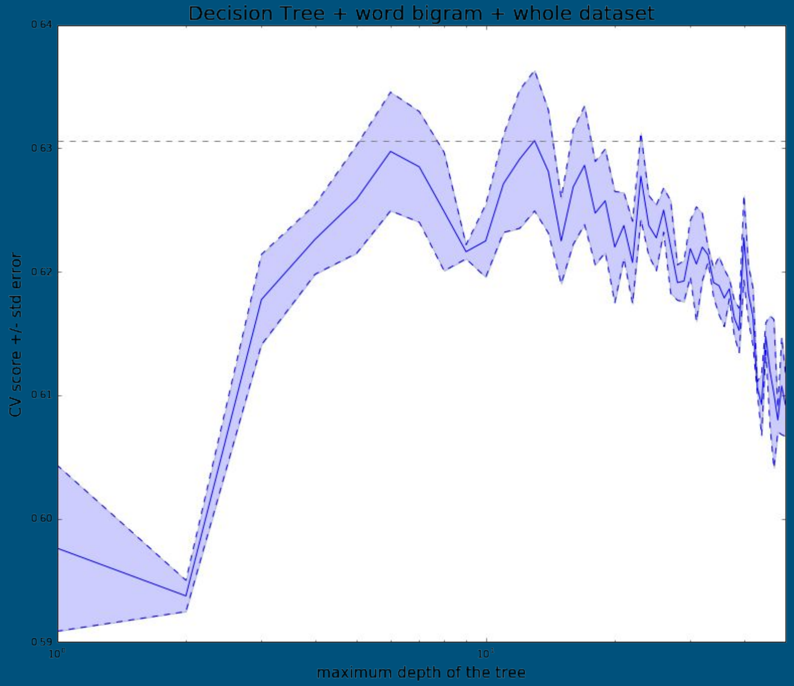
1. Explore parameter in the classifier
    - Maximum depth of the decision tree
    - Overfitting VS Implementing weak predictors
2. Compare balanced vs unbalanced
3. Compare word vs character

Classifiers — Decision Tree

1. Explore parameter in the classifier
2. Balanced dataset VS unbalanced dataset
3. Compare word vs character

Age group	Portion (%)
10s	7
20s	36
30s	57

Classifiers — Decision Tree



dataset	features	Training score	Testing score
balanced	word	0.71	0.64
unbalance	word	0.72	0.65
balanced	character	0.97	0.63

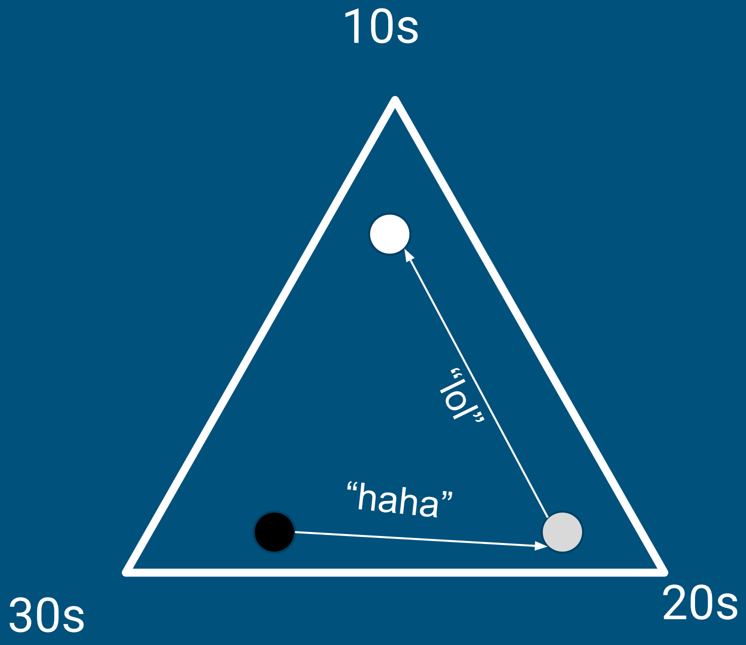
Classifiers — Naive Bayes

Multinomial Naive Bayes

- Multinomial distribution
- Frequency of the features

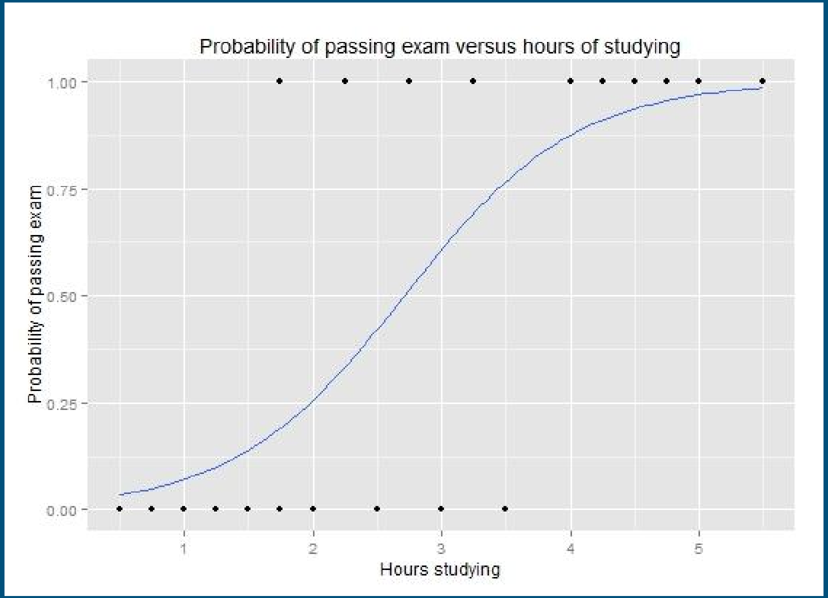
$$\Theta_y = (\Theta_{y1}, \ldots, \Theta_{yn})$$

$$\widehat{\Theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$



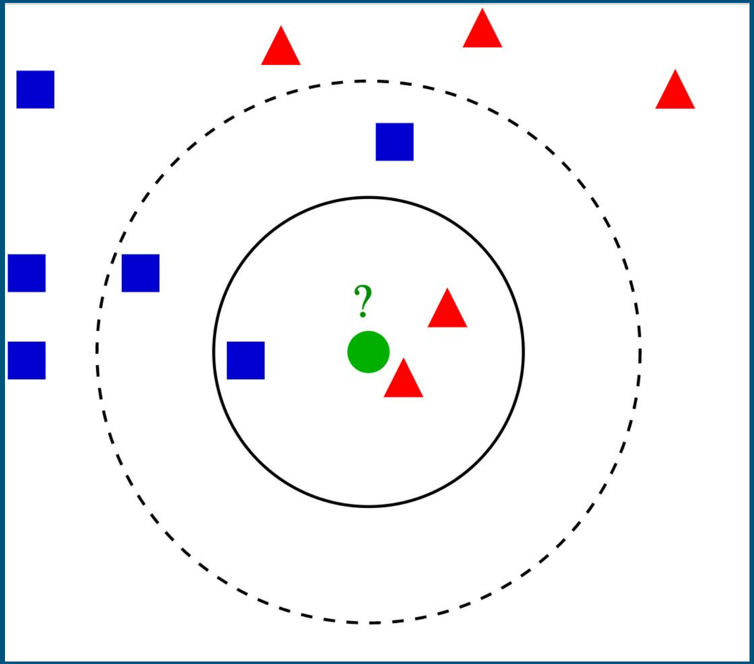
Classifiers — Maximum Entropy

- Uses Logistic Regression to model impact of each feature
- Does not assume statistical independence of features
- Produces coefficients for each possible classification
  - Can be used to integrate with other classifiers.



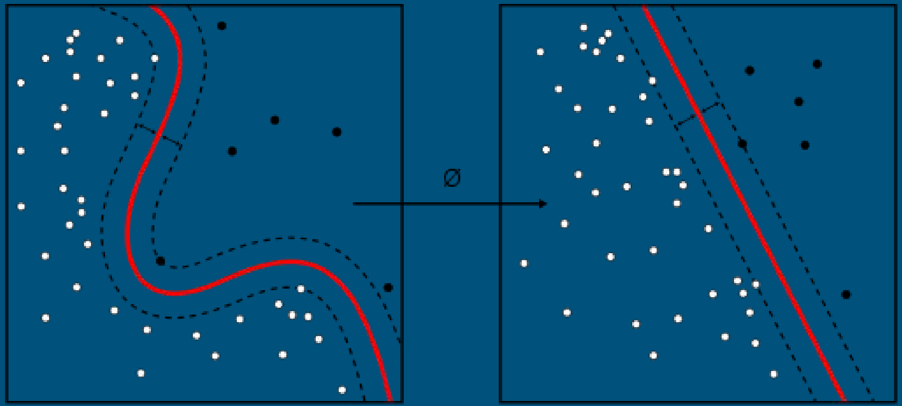
Classifiers — k Nearest Neighbors

- Very simple implementation
- Almost all computation done at classification time
- Very slow for high dimensional data
- Choice of k matters a great deal
- Choice of distance metric matters



Classifiers — Support Vector Machine

- Hyperplane construction to create a margin between features
    - Apply classification and regression
-



Classifiers — Support Vector Machine

Kernel Machine (Similarity Functions)

- Linear Kernel

$$K(x, y) = x^T.y$$

- Polynomial Kernel

$$K(x, y) = (x^T.y + c)^p$$

Results

Accuracies: Word-based Ngram Extraction

Classifier	Age		Gender	
	Training Acc.	Test Acc.	Training Acc.	Test Acc.
kNN	0.755	0.658	0.743	0.607
Decision Trees	0.720	0.650	DNF	DNF
Max Entropy	0.710	0.683	0.682	0.648
SVM (Linear Kernel)	0.735	0.680	0.702	0.631
SVM (Poly Kernel)	0.567	0.569	0.500	0.502

*DNF = Did Not Finish (Took too long to run and did not finish computing accuracies)

Accuracies: Character-based Ngram Extraction

Classifier	Age		Gender	
	Training Acc.	Test Acc.	Training Acc.	Test Acc.
kNN	DNF*	DNF	DNF	DNF
Decision Trees	DNF	DNF	DNF	DNF
Max Entropy	0.756	0.691	0.732	0.666
SVM (Linear Kernel)	0.770	0.532	0.803	0.514
SVM (Poly Kernel)	DNF	DNF	DNF	DNF

*DNF = Did Not Finish (Took too long to run and did not finish computing accuracies)

Conclusions

- Max Entropy and SVM best performance with current tuning
- Character VS Word -based feature extraction has little impact
- Able to detect age much better than a random guess, but not great
  - More data would help

Future Work

- Extract additional types of features such as POS, words per sentence, etc.
- Tune current feature selection method and implement additional methods

Questions?