

# Quantitative text analysis: Machine Learning for Text

Blake Miller

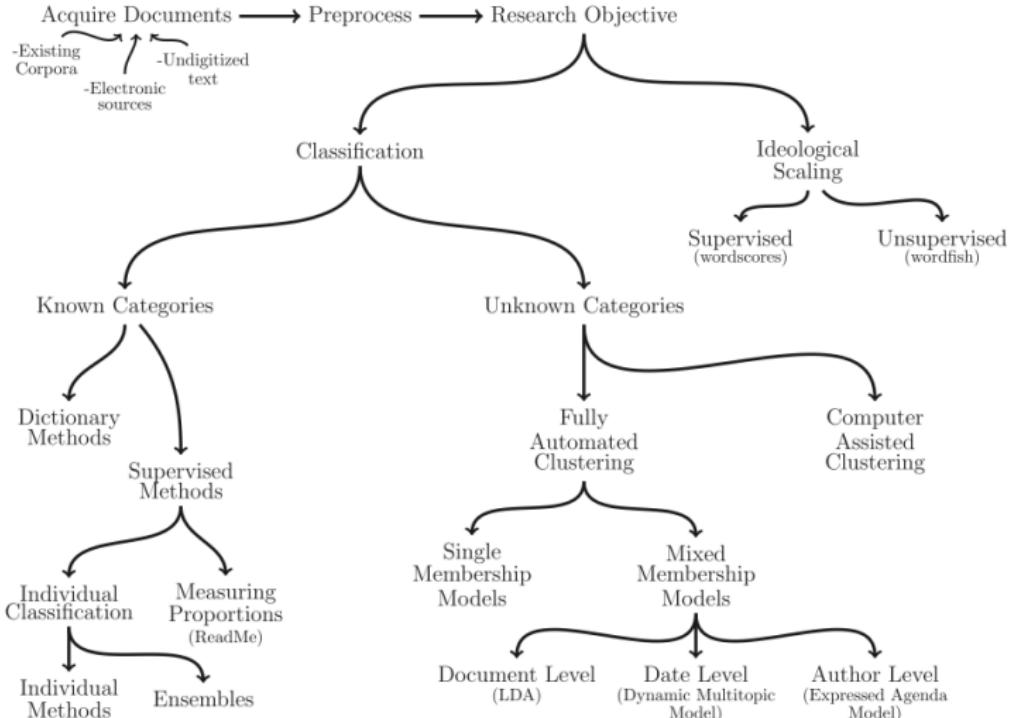
MY 459: Quantitative Text Analysis

February 6, 2023

Course website: [lse-my459.github.io](https://lse-my459.github.io)

1. Overview and Fundamentals
2. Descriptive Statistical Methods for Text Analysis
3. Automated Dictionary Methods
4. Machine Learning for Texts
5. Supervised Scaling Models for Texts
6. *Reading Week*
7. Unsupervised Models for Scaling Texts
8. Similarity and Clustering Methods
9. Topic models
10. Word embeddings
11. Working with Social Media

# Overview of text as data methods



# Outline

- ▶ Supervised learning overview
- ▶ Creating a labeled set and evaluating its reliability
- ▶ Classifier performance metrics
- ▶ Examples of classifiers
  - ▶ Naive Bayes
  - ▶ Regularized regression
- ▶ Active learning for labeling texts
- ▶ Applications of classifiers in social science research

# Supervised machine learning

**Goal:** classify documents into pre existing categories.

e.g. authors of documents, sentiment of tweets, ideological position of parties based on manifestos, tone of movie reviews...

**What we need:**

- ▶ Hand-coded dataset (labeled), to be split into:
  - ▶ **Training set:** used to train the classifier
  - ▶ **Validation/Test set:** used to validate the classifier
- ▶ Method to extrapolate from hand coding to unlabeled documents (**classifier**):
  - ▶ Naive Bayes, regularized regression, SVM, CNN, ensemble methods, etc.
- ▶ Approach to validate classifier: **cross-validation**
- ▶ **Performance metric** to choose best classifier and avoid overfitting: confusion matrix, accuracy, precision, recall...

# Supervised Learning

- ▶  $X$ : data matrix, each column is called a feature, independent variable, input variable (these can be words, n-grams, characters, word embeddings, etc.)
- ▶  $Y$ : dependent variable/target
- ▶ **Objective:** Approximate the function of relationship between  $X$  and  $Y$ , make sure this function generalizes well to new data.
- ▶ **Regression:**  $Y$  is a number
  - ▶ Linear regression is a very specialized case of regression
  - ▶ There are non-linear regression algorithms such as tree regression
- ▶ **Classification:**  $Y$  is a value in an unordered set (i.e.  $y_i \in \{\text{ordinary, bot, troll, cyborg}\}$ )
- ▶ **Training data:** Pairs  $(x_1, y_1), \dots, (x_N, y_N)$  used to "train" a model that will later predict  $y$  when given unseen **test** cases.
  - ▶ What are important/informative features?
  - ▶ How good are the model's predictions?/How useful is the model?

# The Process of Supervised Learning For Text Data

1. Gather documents
2. Preprocess documents
3. Describe the documents
4. Transform the documents (to a dfm)
5. Specify a model
6. Train the model
7. Evaluate model performance

## Classification v. scaling methods compared

- ▶ Machine learning focuses on identifying classes ([classification](#)), while social science is typically interested in locating things on latent traits ([scaling](#))
- ▶ But the two methods overlap and can be adapted – will demonstrate later using the Naive Bayes classifier
- ▶ Applying lessons from machine learning to supervised scaling, we can
  - ▶ Apply classification methods to scaling
  - ▶ Improve it using lessons from machine learning

## Supervised v. unsupervised methods compared

- ▶ The **goal** (in text analysis) is to differentiate *documents* from one another, treating them as “bags of words”
- ▶ Different approaches:
  - ▶ *Supervised methods* for classification require a **training set** that exemplifies contrasting **classes**, identified by the researcher
  - ▶ *Unsupervised methods* identify similarities in documents based on patterns in the term-document matrix, without requiring supervision (human annotations)
- ▶ Relative **advantage** of supervised methods:

You already know the dimension being scaled, because you set it in the training stage
- ▶ Relative **disadvantage** of supervised methods:

You *must* already know the dimension being scaled, because you have to feed it good sample documents in the training stage

## Supervised v. unsupervised methods: Examples

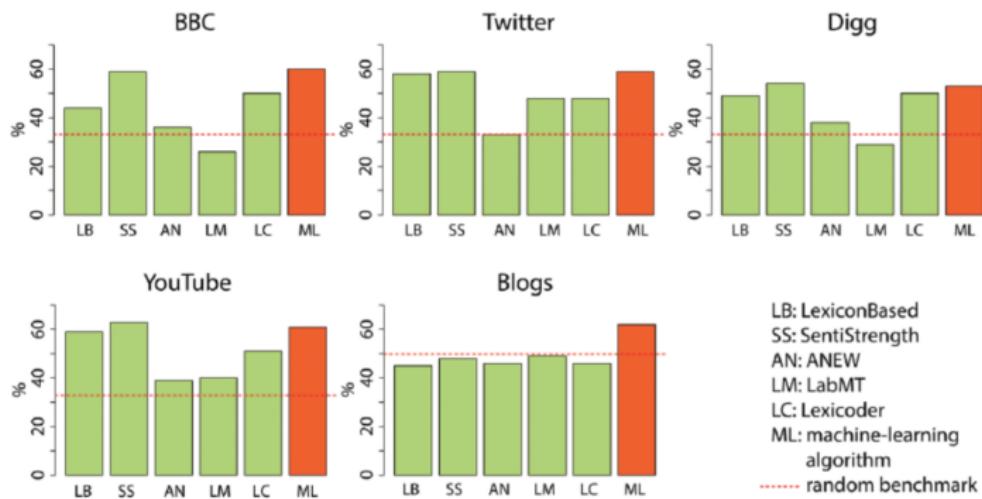
- ▶ General examples:
  - ▶ **Supervised**: Naive Bayes, regularized regression, support vector machines (SVM), convolutional neural networks (CNN)
  - ▶ **Unsupervised**: topic models, IRT models, correspondence analysis, factor analytic approaches
- ▶ Social science applications
  - ▶ **Supervised**: Wordscores (LBG 2003); SVMs (Yu, Kaufman and Diermeier 2008); Naive Bayes (Evans et al 2007)
  - ▶ **Unsupervised**: Structural topic model (Roberts et al 2014); "Wordfish" (Slapin and Proksch 2008); two-dimensional IRT (Monroe and Maeda 2004)

## Supervised learning v. dictionary methods

- ▶ Dictionary methods:
  - ▶ Advantage: **not corpus-specific**, cost to apply to a new corpus is trivial
  - ▶ Disadvantage: **not corpus-specific**, so performance on a new corpus is unknown (domain shift)
- ▶ Supervised learning can be conceptualized as a generalization of dictionary methods, where features associated with each categories (and their relative weight) are **learned from the data**
- ▶ By construction, they will **outperform dictionary methods** in classification tasks, as long as training sample is large enough

# Dictionaries vs supervised learning

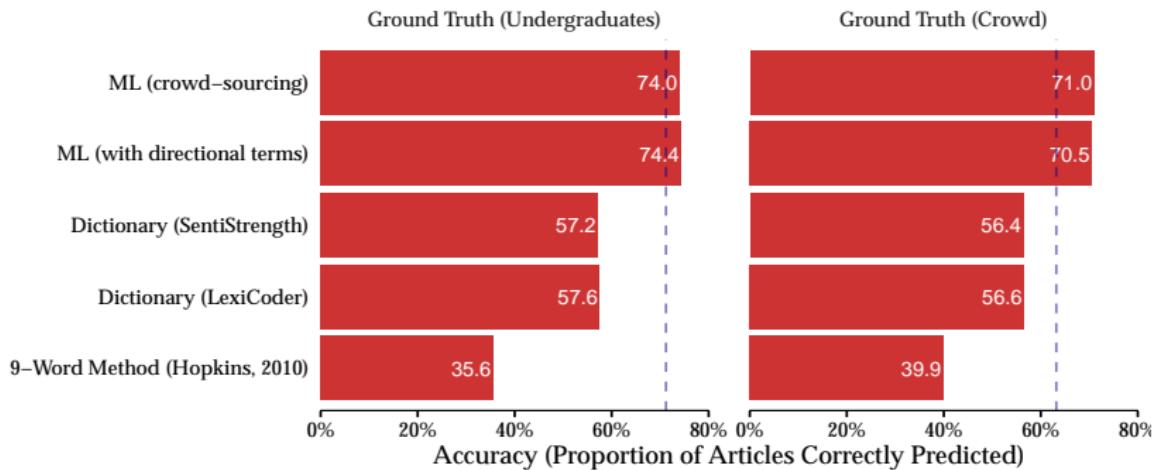
Lexicons' Accuracy in Document Classification  
Compared to Machine-Learning Approach



Source: González-Bailón and Paltoglou (2015)

# Dictionaries vs supervised learning

Application: sentiment analysis of NYTimes articles



**Source:** Barberá et al (2017)

# Outline

- ▶ Supervised learning overview
- ▶ Creating a labeled set and evaluating its reliability
- ▶ Classifier performance metrics
- ▶ Examples of classifiers
  - ▶ Naive Bayes
  - ▶ Regularized regression
- ▶ Active learning for labeling texts
- ▶ Applications of classifiers in social science research

# Creating a labeled set

How do we obtain a **labeled set**?

- ▶ External sources of annotation
  - ▶ Disputed authorship of Federalist papers estimated based on known authors of other documents
  - ▶ Party labels for election manifestos
  - ▶ Legislative proposals by think tanks (text reuse)
- ▶ Expert annotation
  - ▶ “Canonical” dataset in Comparative Manifesto Project
  - ▶ In most projects, undergraduate students (expertise comes from training)
- ▶ Crowd-sourced coding
  - ▶ **Wisdom of crowds:** aggregated judgments of non-experts converge to judgments of experts at much lower cost (Benoit et al, 2016)
  - ▶ Easy to implement with CrowdFlower or MTurk

# Code the Content of a Sample of Tweets

## Instructions ▾

In this job, you will be presented with tweets about the recent protests related to race and law enforcement in the U.S.

You will have to read the tweet and answer a set of questions about its content.

Read the tweet below paying close attention to detail:

Tweet ID: 447



**El Cid**  
@JohnGalt2112

 Follow

#BlackLivesMatter don't matter unless they are  
taken by a white cop.

4:23 PM - 13 Dec 2014

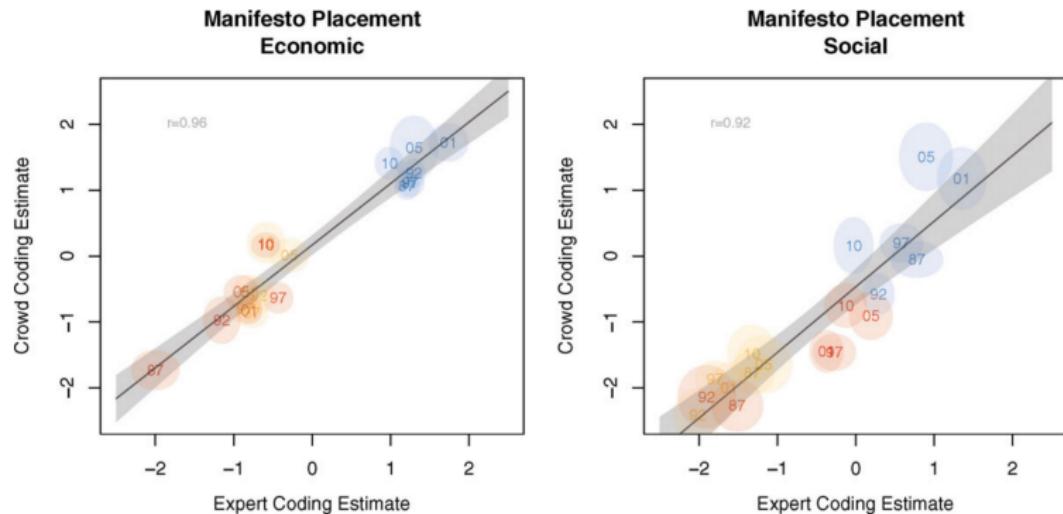


Is this tweet related to the ongoing debate about law enforcement and race in the United States?

- Yes
- No
- Don't Know

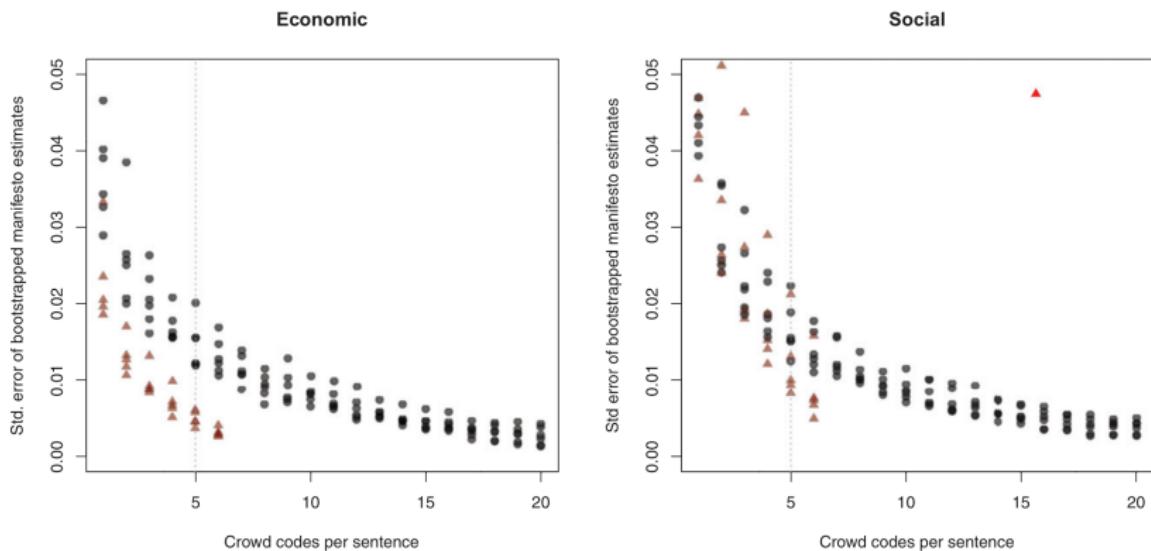
# Crowd-sourced text analysis (Benoit et al, 2016 APSR)

FIGURE 3. Expert and Crowd-sourced Estimates of Economic and Social Policy Positions



# Crowd-sourced text analysis (Benoit et al, 2016 APSR)

**FIGURE 5. Standard Errors of Manifesto-level Policy Estimates as a Function of the Number of Workers, for the Oversampled 1987 and 1997 Manifestos**



*Note:* Each point is the bootstrapped standard deviation of the mean of means aggregate manifesto scores, computed from sentence-level random n subsamples from the codes.

## Evaluating the quality of a labeled set

Any labeled set should be tested and reported for its **inter-rater reliability**, also sometimes called **inter-coder reliability**, at three different standards:

Type	Test Design	Causes of Disagreements	Strength
<b>Stability</b>	test-retest	intraobserver inconsistencies	weakest
<b>Reproducibility</b>	test-test	intraobserver inconsistencies + interobserver disagreements	medium
<b>Accuracy</b>	test-standard	intraobserver inconsistencies + interobserver disagreements + deviations from a standard	strongest

# Measures of agreement

- ▶ Percent agreement Very simple:  
$$(\text{number of agreeing ratings}) / (\text{total ratings}) * 100\%$$
- ▶ Correlation
  - ▶ (usually) Pearson's  $r$ , aka product-moment correlation
  - ▶ Formula: 
$$r_{AB} = \frac{1}{n-1} \sum_{i=1}^n \left( \frac{A_i - \bar{A}}{s_A} \right) \left( \frac{B_i - \bar{B}}{s_B} \right)$$
  - ▶ May also be ordinal, such as Spearman's rho or Kendall's tau-b
  - ▶ Range is  $[0,1]$
- ▶ Agreement measures
  - ▶ Take into account not only observed agreement, but also *agreement that would have occurred by chance*
  - ▶ Cohen's  $\kappa$  is most common
  - ▶ Krippendorff's  $\alpha$  is a generalization of Cohen's  $\kappa$
  - ▶ Both range from  $[0,1]$

## Reliability data matrixes

Example here used binary data (from Krippendorff)

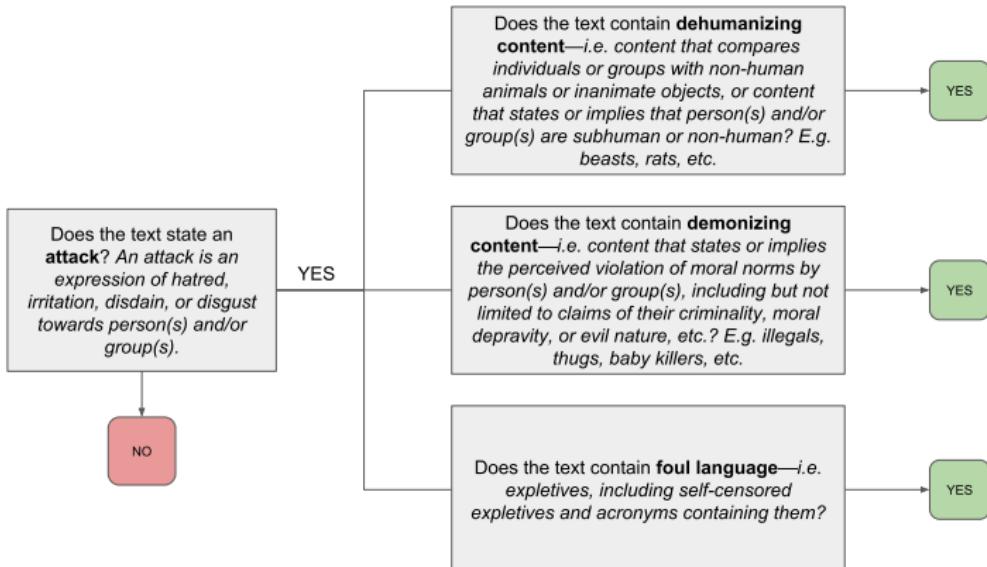
<b>Article:</b>	1	2	3	4	5	6	7	8	9	10
<b>Coder A</b>	1	1	0	0	0	0	0	0	0	0
<b>Coder B</b>	0	1	1	0	0	1	0	1	0	0

- ▶ A and B agree on 60% of the articles: 60% agreement
- ▶ Correlation is (approximately) 0.10
- ▶ Observed *disagreement*: 4
- ▶ Expected *disagreement* (by chance): 4.4211
- ▶ Krippendorff's  $\alpha = 1 - \frac{D_o}{D_e} = 1 - \frac{4}{4.4211} = 0.095$
- ▶ Cohen's  $\kappa$  (nearly) identical

# Example: Identifying Hate Speech

NOTE: The content of an attack doesn't have to be objectively true or false for it to be an attack.

## ATTACK



# Outline

- ▶ Supervised learning overview
- ▶ Creating a labeled set and evaluating its reliability
- ▶ **Classifier performance metrics**
- ▶ Examples of classifiers
  - ▶ Naive Bayes
  - ▶ Regularized regression
- ▶ Active learning for labeling texts
- ▶ Applications of classifiers in social science research

## Basic principles of supervised learning

- ▶ **Generalization:** A classifier or a regression algorithm learns to correctly predict output from given inputs not only in previously seen samples but also in previously unseen samples
- ▶ **Overfitting:** A classifier or a regression algorithm learns to correctly predict output from given inputs in previously seen samples but fails to do so in previously unseen samples. This causes poor prediction/generalization.
- ▶ Goal is to maximize the frontier of precise identification of true condition with accurate recall

## Performance metrics

- ▶ **Accuracy:** How correctly is the classifier's identifications?
  - ▶ % of documents that are correctly predicted.
- ▶ **Precision:** Does the classifier identify *only* my content?
  - ▶ % of documents that are predicted positive that are indeed positive.
- ▶ **Recall:** Does the classifier identify *all* my content?
  - ▶ % of positive documents that are predicted positive.

## Performance metrics: confusion matrix

		True condition	
		Positive	Negative
Prediction	Positive	True Positive	False Positive (Type I error)
	Negative	False Negative (Type II error)	True Negative

- ▶ Accuracy:  $\frac{\text{Correctly classified}}{\text{Total number of cases}} = \frac{\text{true positives} + \text{true negatives}}{\text{Total number of cases}}$
- ▶ Precision:  $\frac{\text{true positives}}{\text{true positives} + \text{false positives}}$
- ▶ Recall:  $\frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$
- ▶  $F1 \text{ score} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$   
(the harmonic mean of precision and recall)

## Example: measuring performance

Assume:

- ▶ We have a corpus where 80 documents are really positive (as opposed to negative, as in sentiment)
- ▶ Our method declares that 60 are positive
- ▶ Of the 60 declared positive, 45 are actually positive

Solution:

$$\text{Precision} = (45 / (45 + 15)) = 45 / 60 = 0.75$$

$$\text{Recall} = (45 / (45 + 35)) = 45 / 80 = 0.56$$

# Accuracy?

		True condition		
		Positive	Negative	
Prediction	Positive	45	60	
	Negative	80		
				105

add in the cells we can compute

		True condition		
		Positive	Negative	
Prediction	Positive	45	15	60
	Negative	35		
		80		

but need True Negatives and  $N$  to compute accuracy

		True condition		60
		Positive	Negative	
Prediction	Positive	45	15	60
	Negative	35	???	
		80		

assume 10 True Negatives:

		True condition		
		Positive	Negative	
Prediction	Positive	45	15	60
	Negative	35	10	45
		80	25	105

$$\text{Accuracy} = (45 + 10)/105 = 0.52$$

$$F1 = 2 * (0.75 * 0.56) / (0.75 + 0.56) = 0.64$$

now assume 100 True Negatives:

		True condition		
		Positive	Negative	
Prediction	Positive	45	15	60
	Negative	35	100	135
		80	115	195

$$\text{Accuracy} = (45 + 100)/195 = 0.74$$

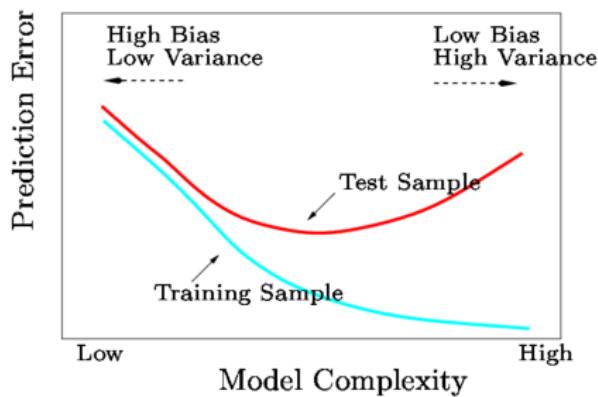
$$F1 = 2 * (0.75 * 0.56) / (0.75 + 0.56) = 0.64$$

## Measuring performance

- ▶ Precision and recall can be reported separately for each category
- ▶ Precision and recall (or F1) should be reported alongside accuracy. [Why?](#)
- ▶ There is generally a trade-off between precision and recall. [Why?](#)

# Measuring performance

- ▶ Classifier is trained to maximize in-sample performance
- ▶ But generally we want to apply method to new data
- ▶ Danger: overfitting



- ▶ Model is too complex, describes noise rather than signal (Bias-Variance trade-off)
- ▶ Focus on features that perform well in labeled data but may not generalize (e.g. "inflation" in 1980s)
- ▶ In-sample performance better than out-of-sample performance

- ▶ Solutions?
  - ▶ Randomly split dataset into training and test set
  - ▶ Cross-validation

# Cross-validation

Intuition:

- ▶ Create K training and test sets (“folds”) within training set.
- ▶ For each k in K, run classifier and estimate performance in test set within fold.
- ▶ Choose best classifier based on cross-validated performance



# Outline

- ▶ Supervised learning overview
- ▶ Creating a labeled set and evaluating its reliability
- ▶ Classifier performance metrics
- ▶ Examples of classifiers
  - ▶ Naive Bayes
  - ▶ Regularized regression
- ▶ Active learning for labeling texts
- ▶ Applications of classifiers in social science research

## Multinomial Bayes model of Class given a Word

Consider  $J$  word types distributed across  $N$  documents, each assigned one of  $K$  classes.

*At the word level*, Bayes Theorem tells us that:

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j)}$$

For two classes, this can be expressed as

$$= \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})} \quad (1)$$

## Multinomial Bayes model of Class given a Word Class-conditional word likelihoods

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})}$$

- ▶ The word likelihood within class
- ▶ The maximum likelihood estimate is simply the proportion of times that word  $j$  occurs in class  $k$ , but it is more common to use Laplace smoothing by adding 1 to each observed count within class

## Multinomial Bayes model of Class given a Word Word probabilities

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j)}$$

- ▶ This represents the **word probability** from the training corpus
- ▶ Usually uninteresting, since it is constant for the training data; is independent of the class  $k$ .
- ▶ This is called a **scaling factor** and is only needed to compute posteriors on a probability scale.

## Multinomial Bayes model of Class given a Word Class prior probabilities

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})}$$

- ▶ This represents the class prior probability
- ▶ Machine learning typically takes this as the document frequency in the training set

## Multinomial Bayes model of Class given a Word Class posterior probabilities

$$P(c_k|w_j) = \frac{P(w_j|c_k)P(c_k)}{P(w_j|c_k)P(c_k) + P(w_j|c_{\neg k})P(c_{\neg k})}$$

- ▶ This represents the **posterior probability of membership in class  $k$**  for word  $j$
- ▶ Key for the classifier: in new documents, we only observe word distributions and want to predict class

## Moving to the document level

- ▶ The “Naive” Bayes model of a joint document-level class posterior **assumes conditional independence**, to multiply the word likelihoods from a “test” document, to produce:

$$P(c_k|d) = P(c_k) \prod_j^J \frac{P(w_j|c_k)}{P(w_j)}$$
$$P(c_k|d) \propto P(c_k) \prod_j^J P(w_j|c_k)$$

- ▶ This is why we call it “naive”: because it (wrongly) assumes:
  - ▶ **conditional independence** of word counts given the class.
    - ▶ If the word “weather” appears in a document, it is more likely that we will see words like “forecast” and “report”
  - ▶ **positional independence** of word counts (bag of words assumption)
    - ▶ It is meaningful for words like “not” and “good” to be close together.
    - ▶ “Peace, no more war”, “War, no more peace” (Source: Arthur Spirling)

# Naive Bayes Classification Example

(From Manning, Raghavan and Schütze, *Introduction to Information Retrieval*)

► **Table 13.1** Data for parameter estimation examples.

	docID	words in document	in $c = \text{China?}$
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

# Naive Bayes Classification Example

**Example 13.1:** For the example in Table 13.1, the multinomial parameters we need to classify the test document are the priors  $\hat{P}(c) = 3/4$  and  $\hat{P}(\bar{c}) = 1/4$  and the following conditional probabilities:

$$\begin{aligned}\hat{P}(\text{Chinese}|c) &= (5+1)/(8+6) = 6/14 = 3/7 \\ \hat{P}(\text{Tokyo}|c) = \hat{P}(\text{Japan}|c) &= (0+1)/(8+6) = 1/14 \\ \hat{P}(\text{Chinese}|\bar{c}) &= (1+1)/(3+6) = 2/9 \\ \hat{P}(\text{Tokyo}|\bar{c}) = \hat{P}(\text{Japan}|\bar{c}) &= (1+1)/(3+6) = 2/9\end{aligned}$$

The denominators are  $(8+6)$  and  $(3+6)$  because the lengths of  $text_c$  and  $text_{\bar{c}}$  are 8 and 3, respectively, and because the constant  $B$  in Equation (13.7) is 6 as the vocabulary consists of six terms.

We then get:

$$\begin{aligned}\hat{P}(c|d_5) &\propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003. \\ \hat{P}(\bar{c}|d_5) &\propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001.\end{aligned}$$

Thus, the classifier assigns the test document to  $c = China$ . The reason for this classification decision is that the three occurrences of the positive indicator Chinese in  $d_5$  outweigh the occurrences of the two negative indicators Japan and Tokyo.

# Outline

- ▶ Supervised learning overview
- ▶ Creating a labeled set and evaluating its reliability
- ▶ Classifier performance metrics
- ▶ Examples of classifiers
  - ▶ Naive Bayes
  - ▶ Regularized regression
- ▶ Active learning for labeling texts
- ▶ Applications of classifiers in social science research

# Regularized regression

Assume we have:

- ▶  $i = 1, 2, \dots, N$  documents
- ▶ Each document  $i$  is in class  $y_i = 0$  or  $y_i = 1$
- ▶  $j = 1, 2, \dots, J$  unique features
- ▶ And  $x_{ij}$  as the count of feature  $j$  in document  $i$

We could build a linear regression model as a classifier, using the values of  $\beta_0, \beta_1, \dots, \beta_J$  that minimize:

$$RSS = \sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^J \beta_j x_{ij} \right)^2$$

But can we?

- ▶ If  $J > N$ , OLS does not have a unique solution
- ▶ Even with  $N > J$ , OLS has low bias/high variance (**overfitting**)

## Regularized regression

What can we do? Add a **penalty for model complexity**, such that we now minimize:

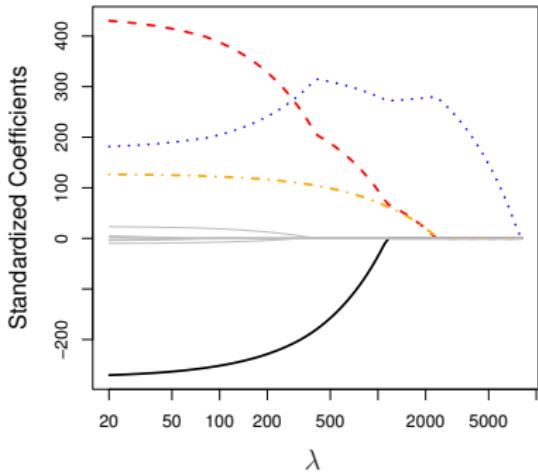
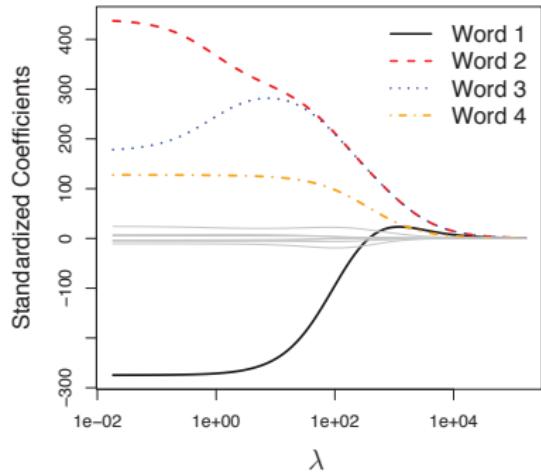
$$\sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^J \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^J \beta_j^2 \rightarrow \text{ridge regression}$$

or

$$\sum_{i=1}^N \left( y_i - \beta_0 - \sum_{j=1}^J \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^J |\beta_j| \rightarrow \text{lasso regression}$$

where  $\lambda$  is the **penalty parameter** (to be estimated)

# Visualizing Lasso vs. Ridge Regression



- ▶ We penalize large coefficients more as the size of  $\lambda$  increases
- ▶ Ridge regression shrinks some parameters close to zero but never quite gets there.
- ▶ Lasso shrinks some parameters to exactly zero.

# Regularized regression

Why the penalty (shrinkage)?

- ▶ Reduces the variance
- ▶ Identifies the model if  $J > N$
- ▶ Some coefficients become zero (feature selection)

The penalty can take different forms:

- ▶ Ridge regression:  $\lambda \sum_{j=1}^J \beta_j^2$  with  $\lambda > 0$ ; and when  $\lambda = 0$  becomes OLS
- ▶ Lasso  $\lambda \sum_{j=1}^J |\beta_j|$  where some coefficients become zero.
- ▶ Elastic Net:  $\lambda_1 \sum_{j=1}^J \beta_j^2 + \lambda_2 \sum_{j=1}^J |\beta_j|$  (best of both worlds?)

How to find best value of  $\lambda$ ? Cross-validation.

Evaluation: regularized regression is easy to interpret, but often outperformed by more complex methods.

# Outline

- ▶ Supervised learning overview
- ▶ Creating a labeled set and evaluating its reliability
- ▶ Classifier performance metrics
- ▶ Examples of classifiers
  - ▶ Naive Bayes
  - ▶ Regularized regression
- ▶ Active learning for labeling texts
- ▶ Applications of classifiers in social science research

## Motivating Example: Government Astroturfer Detection

### The Fifty Cent Party



"Netizens first coined the term 'Fifty Cent Party' to refer to undercover Internet commentators paid by the government to sway public opinion ('fifty cents' is a reference to the alleged pay received per post)."

—China Digital Times

# Motivating Example: Government Astroturfer Detection

Amazon review astroturfing exhibit from the Amazon v. Gentile lawsuit in Washington Superior Court.

★★★★★ **Cool charger**

By [Tiffany](#) on March 30, 2015

Verified Purchase

Bought this for my Galaxy phone and I have to say, this is a pretty cool USB cord! :) I like the lights in the cord as it puts off a cool glowing effect in my room at night and it makes it much easier to see, thanks for the great product!

★★★★★ **Definitely buying more.**

By [Krystal Willingham](#) on March 28, 2015

Verified Purchase

I was impressed with how bright the lights on the cable are. It works amazing and as described. i received earlier than expected so that made me very happy. So far is working like a charm and I can't wait to buy a few more.

- ▶ *Research Challenge:* Need to disambiguate **government** astroturfers and **non-government** astroturfers
- ▶ *Potential Solution:* Use information from the social network of users. Do they follow **government accounts**?

# Motivating Example: Government Astroturfer Detection

Local government leaders are evaluated for promotion based on their online influence.

Rank	Weibo Name	Weibo Description/Affiliation	Message Reach	Public Service	Inter-activity	Public Acceptance	Overall Score
1	Xinjiang Earthquake Administration	Official Weibo Account of the Xinjiang Earthquake Administration	74.11	74.78	74.02	78.99	74.84
2	Rapid Road Traffic Police	Official Weibo Account of the Urumqi City Rapid Road Traffic Police	72.75	81.73	58.03	59.14	70.56
3	Peaceful Shihezi	Official Weibo Account of the Shihezi City Public Security Bureau	58.97	86.68	53.83	54.19	68.04
4	Xinjiang Railways	Official Weibo Account of the Urumqi Railway Administration	63.94	81.00	47.14	51.89	64.52
5	Altay Online Public Security	Official Weibo Account of the Xinjiang Uighur Autonomous Region Altay Public Security Bureau	65.44	57.62	69.06	70.90	63.95
6	Xinjiang Anti-Cult	Official Weibo of the Office of the Leading Group for the Prevention and Treatment of Cults in Xinjiang Uygur Autonomous Region	62.44	51.90	64.17	60.95	60.70
7	Peaceful Xinjiang Online	Official Weibo Account of the Xinjiang Provincial Public Security Bureau	48.29	66.99	57.78	52.21	55.27
8	Xinjiang Fire Corps	Official Weibo Account of the Xinjiang Fire Corps	55.98	60.89	49.51	39.97	54.40
9	Hetian Internet Police Inspection and Law Enforcement	Official Weibo Account of the Xinjiang Hetian District Public Security Bureau Network Security Detachment	56.68	61.78	48.83	27.95	53.49
10	Changji Fire Brigade	Official Weibo Account of the Xinjiang Changji Prefecture Public Security Fire Brigade	55.71	60.03	45.57	43.96	53.22

Central government rankings of Weibo accounts in Xinjiang Province

# Motivating Example: Government Astroturfer Detection

Bureaucrats are often required to follow the account of the Document from a local propaganda department email leak.

Zhanggong District Dept. of Education  
Followers of Zhanggong Propaganda Department Weibo Account

Index	School (Work Unit)	Name	Weibo Username	Have they followed?
1	Ganzhou No.7 Middle School	[REDACTED]	[REDACTED]	Yes
2	Ganzhou No.7 Middle School	[REDACTED]	[REDACTED]	Yes
3	Ganzhou No.7 Middle School	[REDACTED]	[REDACTED]	Yes
4	Ganzhou No.7 Middle School	[REDACTED]	[REDACTED]	Yes
5	Ganzhou No.7 Middle School	[REDACTED]	[REDACTED]	Yes
6	Ganzhou Cuo'e Temple Elementary School	[REDACTED]	[REDACTED]	Yes

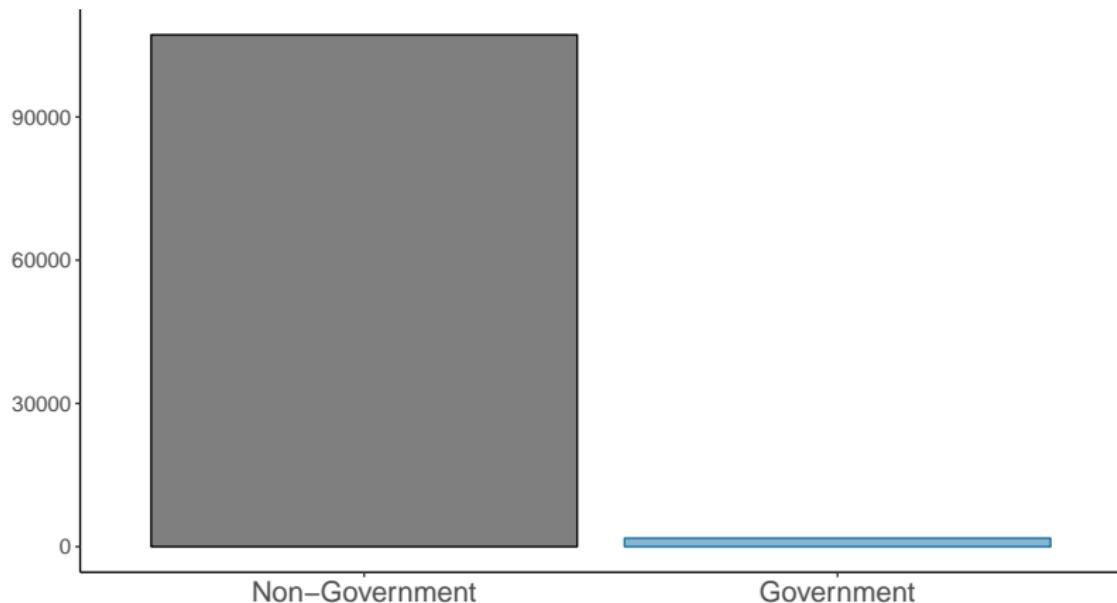
Document from a local propaganda department email leak.

## Motivating Example: Classifying Government Accounts

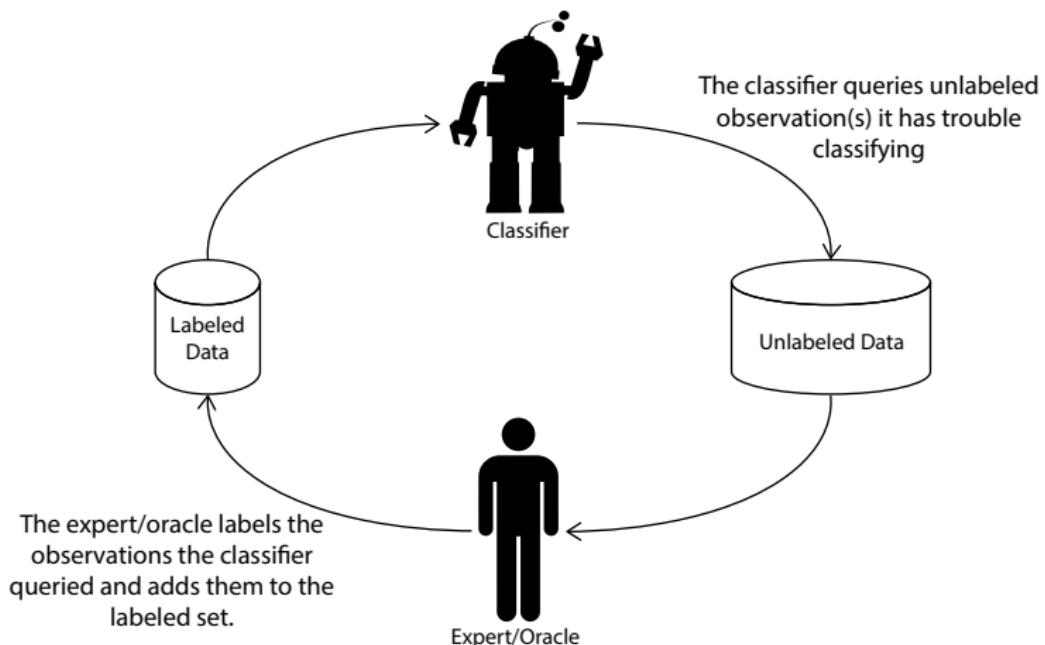
- ▶ Need to automatically **classify** Weibo accounts as government and non-government.
- ▶ Build a **training set** of Weibo accounts using **active learning**.
- ▶ Use this training sample to classify all followed accounts in a corpus of 80 million news comments.

## Motivating Example: Classifying Government Accounts

One major concern: the two classes for this problem are highly imbalanced.



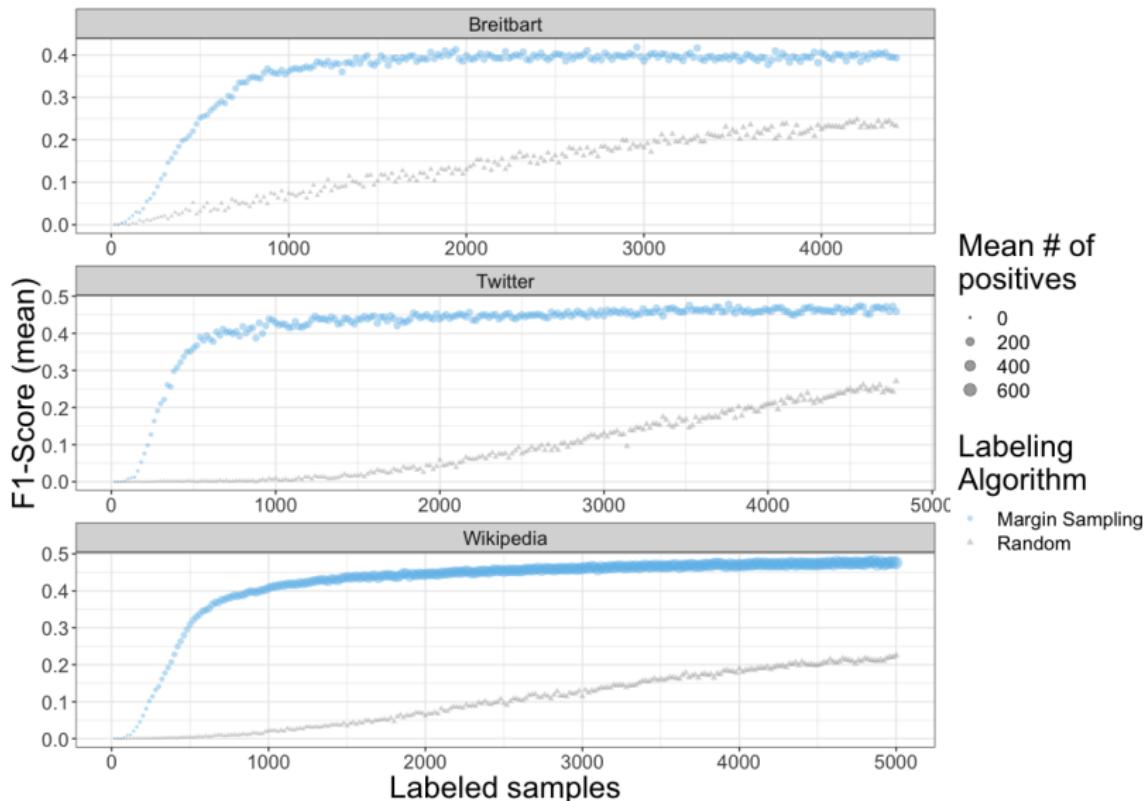
# A Potential Solution: Active Learning



## Active Learning (details)

- ▶ For supervised learning, we need human labels for a subset of documents.
  - ▶ This can be expensive and time consuming, especially when **classes are imbalanced** (e.g. government vs. non-government social media accounts)
- ▶ Active learning instead samples observation based on an **uncertainty measure** derived from one or more classifiers.
  - ▶ Logic: Rather than wasting effort on easily classifiable observations, focus on the hard cases.
  - ▶ Many potential **uncertainty measures** (e.g. observations with  $p \approx .5$  for a logit model)
- ▶ Active learning can dramatically decrease the number of observations we must label for supervised learning (Miller et. al., 2020)
- ▶ **Caveat:** Error measures will be biased; should use randomly sampled test set to compute error.

# Comparing Active and Passive Learning



Source: Miller et. al. (2020)

# Passive Learning

	Party	government	good	football	China	...
X <sub>1</sub>	8	9	5	0	12	...
X <sub>2</sub>	0	0	8	4	0	...
X <sub>3</sub>	0	0	7	4	2	...
X <sub>4</sub>	9	8	6	0	8	...
...	...	...	...	...	...	...

Random Sample



Expert/Oracle

# Passive Learning

	Party	government	good	football	China	...
X <sub>1</sub>	8	9	5	0	12	...
X <sub>2</sub>	0	0	8	4	0	...
X <sub>3</sub>	0	0	7	4	2	...
X <sub>4</sub>	9	8	6	0	8	...
...	...	...	...	...	...	...

Random Sample

x	y
X <sub>1</sub>	GOV
X <sub>2</sub>	¬GOV
X <sub>3</sub>	¬GOV
X <sub>4</sub>	GOV
...	...

An expert labels  
each account as  
**government** or  
**non-government**



Expert/Oracle

# Passive Learning

	Party	government	good	football	China	...
X <sub>1</sub>	8	9	5	0	12	...
X <sub>2</sub>	0	0	8	4	0	...
X <sub>3</sub>	0	0	7	4	2	...
X <sub>4</sub>	9	8	6	0	8	...
...	...	...	...	...	...	...

Random Sample



Classifier

Train a classifier  
on labeled documents

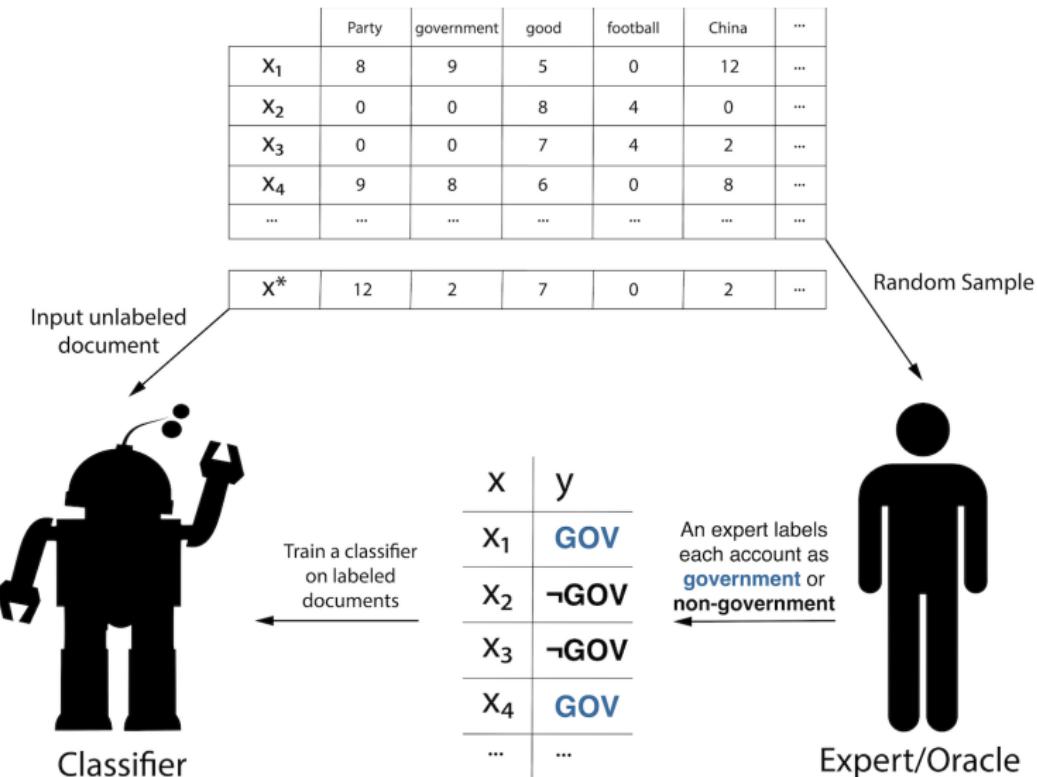
x	y
X <sub>1</sub>	GOV
X <sub>2</sub>	¬GOV
X <sub>3</sub>	¬GOV
X <sub>4</sub>	GOV
...	...

An expert labels  
each account as  
**government** or  
non-government

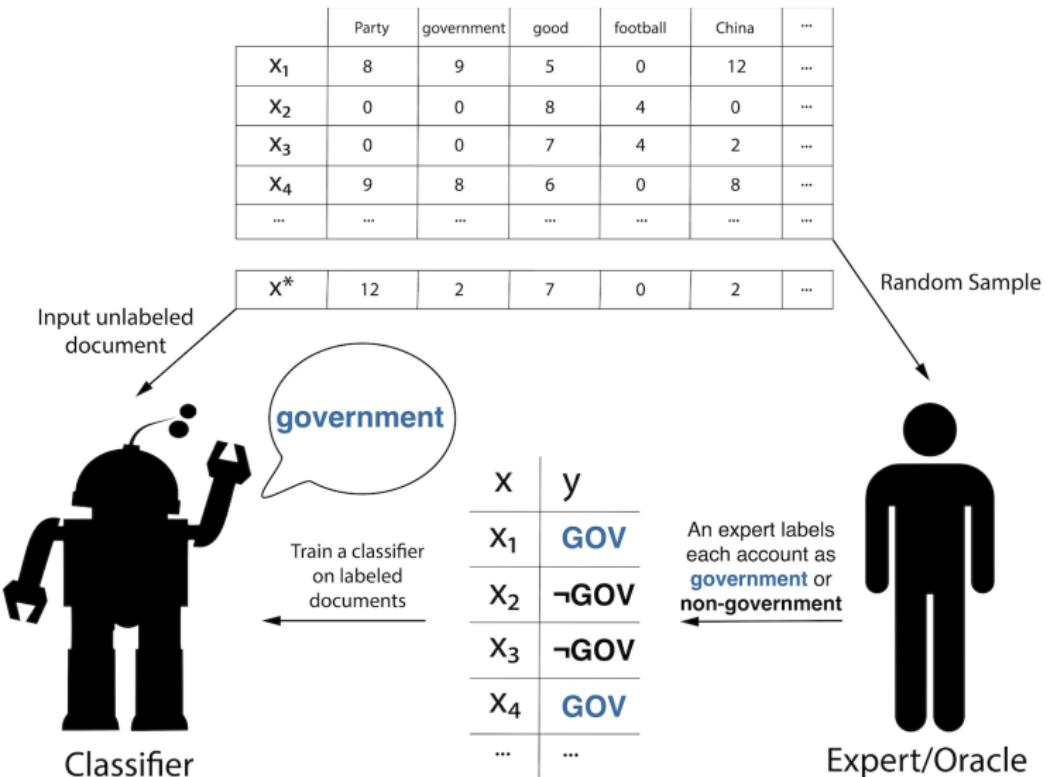


Expert/Oracle

# Passive Learning



# Passive Learning



# Active Learning

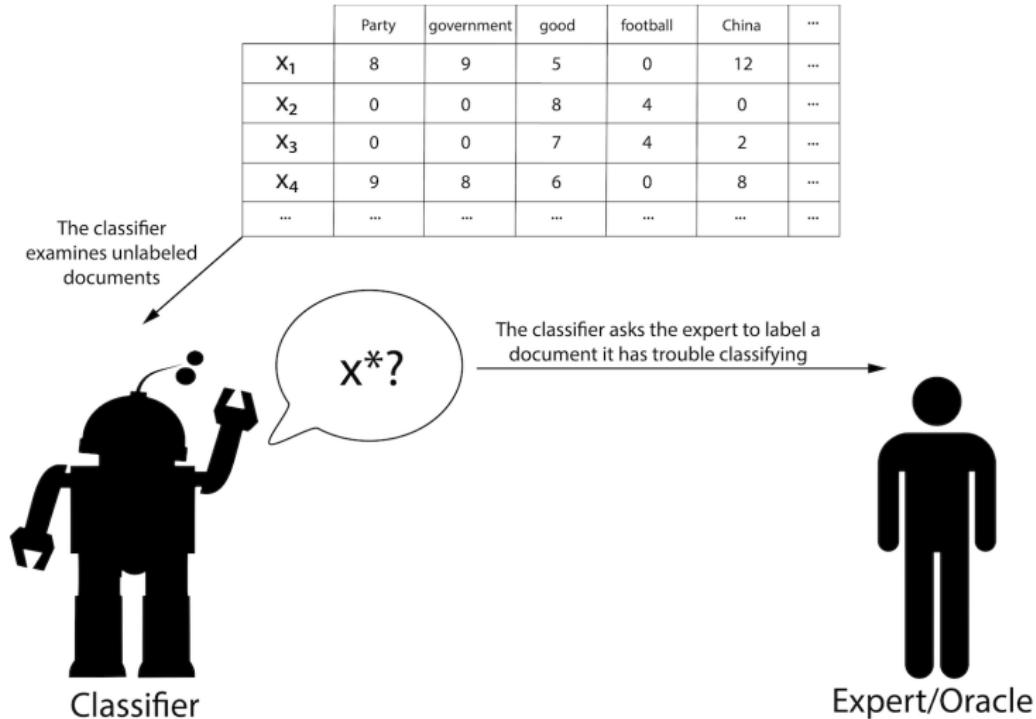
	Party	government	good	football	China	...
X <sub>1</sub>	8	9	5	0	12	...
X <sub>2</sub>	0	0	8	4	0	...
X <sub>3</sub>	0	0	7	4	2	...
X <sub>4</sub>	9	8	6	0	8	...
...	...	...	...	...	...	...

The classifier  
examines unlabeled  
documents

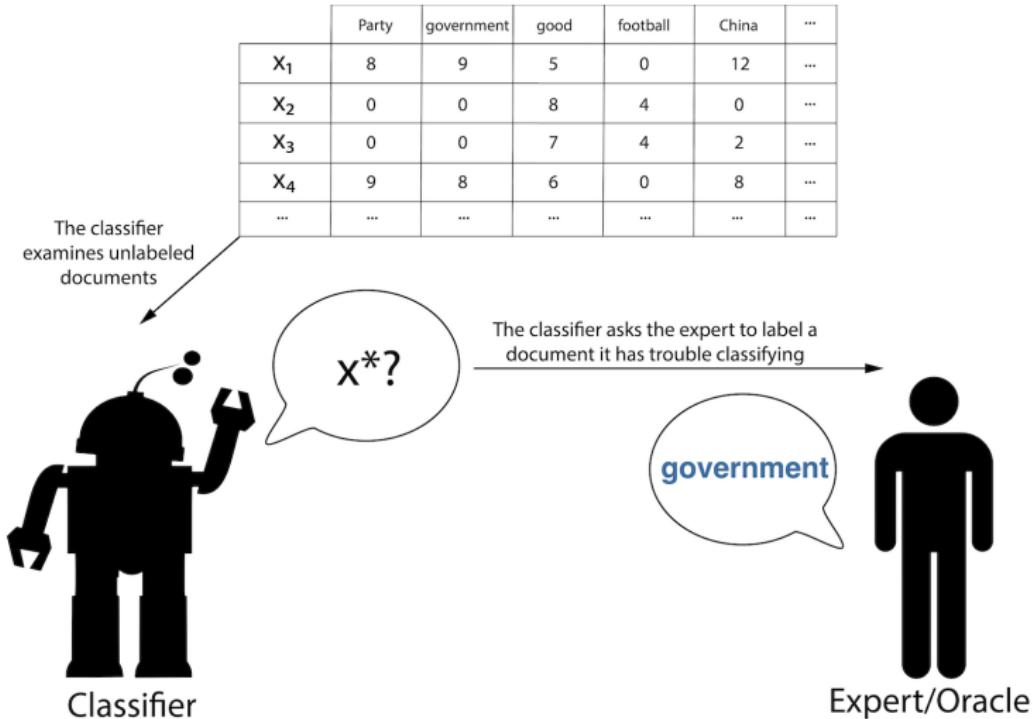


Classifier

# Active Learning



# Active Learning



## Example: Classifying Government Accounts

Model performance (F1): 0.93

## Avatars of predicted government accounts.



# Outline

- ▶ Supervised learning overview
- ▶ Creating a labeled set and evaluating its reliability
- ▶ Classifier performance metrics
- ▶ Examples of classifiers
  - ▶ Naive Bayes
  - ▶ Regularized regression
- ▶ Active learning for labeling texts
- ▶ Applications of classifiers in social science research

## Example: Theocharis et al (2016 JOC)

Why do politicians not take full advantage of interactive affordances of social media?

A politician's incentive structure

Democracy → Dialogue > Mobilisation > Marketing

Politician → Marketing > Mobilisation > Dialogue\*

H1: Politicians make broadcasting rather than engaging use of Twitter

H2: Engaging style of tweeting is positively related to impolite or uncivil responses

## Data collection and case selection

**Data:** European Election Study 2014, Social Media Study

- ▶ List of all candidates with Twitter accounts in 28 EU countries
  - ▶ 2,482 out of 15,527 identified MEP candidates (16%)
- ▶ Collaboration with TNS Opinion to collect all tweets by candidates *and* tweets mentioning candidates (tweets, retweets, @-replies), May 5th to June 1st 2014.

**Case selection:** expected variation in politeness/civility

	Received bailout	Did not receive bailout
High support for EU	Spain (55.4%)	Germany (68.5%)
Low support for EU	Greece (43.8%)	UK (41.4%)

(% indicate proportion of country that considers the EU to be “a good thing”)

## Data collection and case selection

Data coverage by country

Country	Lists	Candidates	on Twitter	Tweets
Germany	9	501	123 (25%)	86,777
Greece	9	359	99 (28%)	18,709
Spain	11	648	221 (34%)	463,937
UK	28	733	304 (41%)	273,886

# Coding tweets

**Coded data:** random sample of ~7,000 tweets from each country, labeled by undergraduate students:

## 1. Politeness

- ▶ Polite: tweet adheres to politeness standards.
- ▶ Impolite: ill-mannered, disrespectful, offensive language...

## 2. Communication style

- ▶ Broadcasting: statement, expression of opinion
- ▶ Engaging: directed to someone else/another user

## 3. Political content: moral and democracy

- ▶ Tweets make reference to: freedom and human rights, traditional morality, law and order, social harmony, democracy...

**Incivility** = impoliteness + moral and democracy

## Coding tweets

### Coding process: summary statistics

	Germany	Greece	Spain	UK
Coded by 1/by 2	2947/2819	2787/2955	3490/1952	3189/3296
Total coded	5766	5742	5442	6485
Impolite	399	1050	121	328
Polite	5367	4692	5321	6157
% Agreement	92	80	93	95
Krippendorf/Maxwell	0.30/0.85	0.26/0.60	0.17/0.87	0.54/0.90
Broadcasting	2755	2883	1771	1557
Engaging	3011	2859	3671	4928
% Agreement	79	85	84	85
Krippendorf/Maxwell	0.58/0.59	0.70/0.70	0.66/0.69	0.62/0.70
Moral/Dem.	265	204	437	531
Other	5501	5538	5005	5954
% Agreement	95	97	96	90
Krippendorf/Maxwell	0.50/0.91	0.53/0.93	0.41/0.92	0.39/0.81

# Machine learning classification of tweets

Coded tweets as training dataset for a machine learning classifier:

1. **Text preprocessing:** lowercase, remove stopwords and punctuation (except # and @), transliterating to ASCII, stem, tokenize into unigrams and bigrams. Keep tokens in 2+ tweets but <90%.
2. **Train classifier:** logistic regression with L2 regularization (ridge regression), one per language and variable
3. **Evaluate classifier:** compute accuracy using 5-fold crossvalidation

## Machine learning classification of tweets

Classifier performance (5-fold cross-validation)

		UK	Spain	Greece	Germany
Communication Style	Accuracy	0.821	0.775	0.863	0.806
	Precision	0.837	0.795	0.838	0.818
	Recall	0.946	0.890	0.894	0.832
Polite vs. impolite	Accuracy	0.954	0.976	0.821	0.935
	Precision	0.955	0.977	0.849	0.938
	Recall	0.998	1.000	0.953	0.997
Morality and Democracy	Accuracy	0.895	0.913	0.957	0.922
	Precision	0.734	0.665	0.851	0.770
	Recall	0.206	0.166	0.080	0.061

## Top predictive n-grams

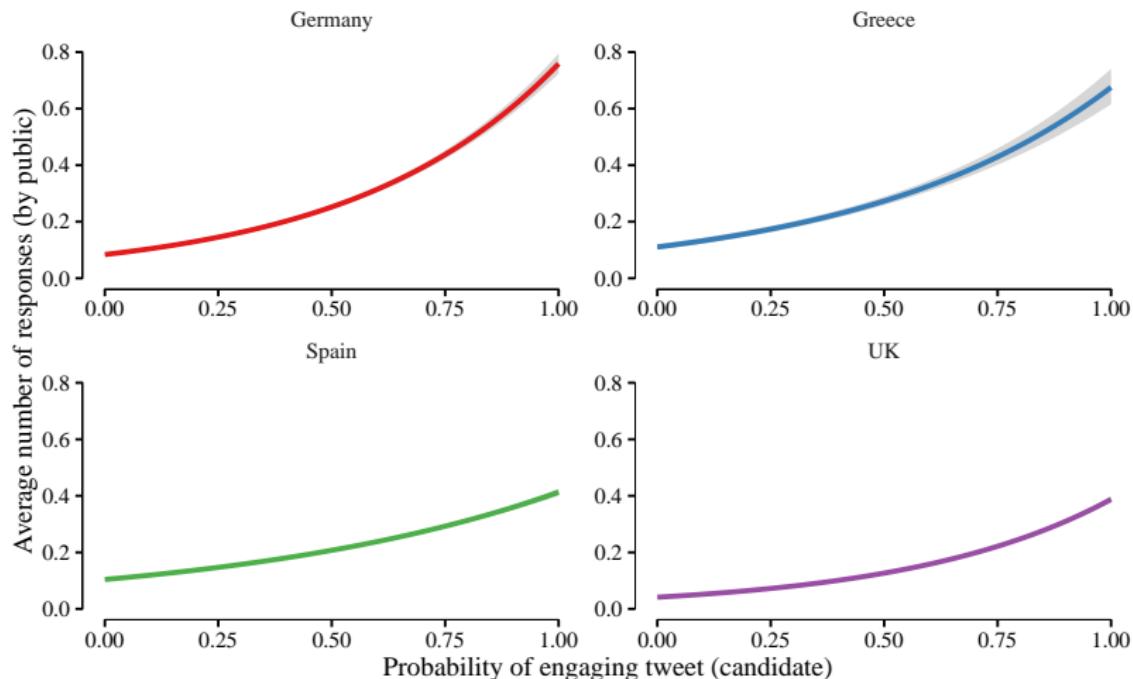
---

Broadcasting	just, hack, #votegreen2014, :, and, @ ', tonight, candid, up, tonbridg, vote @, im @, follow ukip, ukip @, #telleurop, angri, #ep2014, password, stori, #vote2014, team, #labourdoorstep, crimin, bbc news
Engaging	@ thank, @ ye, you'r, @ it', @ mani, @ pleas, u, @ hi, @ congratul, :), index, vote # skip, @ good, fear, cheer, haven't, lol, @ i've, you've, @ that', choice, @ wa, @ who, @ hope
Impolite	cunt, fuck, twat, stupid, shit, dick, tit, wanker, scumbag, moron, cock, foot, racist, fascist, sicken, fart, @ fuck, ars, suck, nigga, nigga ?, smug, idiot, @arsehol, arsehol
Polite	@ thank, eu, #ep2014, thank, know, candid, veri, politician, today, way, differ, europ, democraci, interview, time, tonight, @ think, news, european, sorri, congratul, good, :, democrat, seat
Moral/Dem.	democraci, polic, freedom, media, racist, gay, peac, fraud, discrimin, homosexu, muslim, equal, right, crime, law, violenc, constitut, faith, bbc, christian, marriag, god, cp, racism, sexist
Others	@ ha, 2, snp, nice, tell, eu, congratul, campaign, leav, alreadi, wonder, vote @, ;), hust, nh, brit, tori, deliv, bad, immigr, #ukip, live, count, got, roma

---

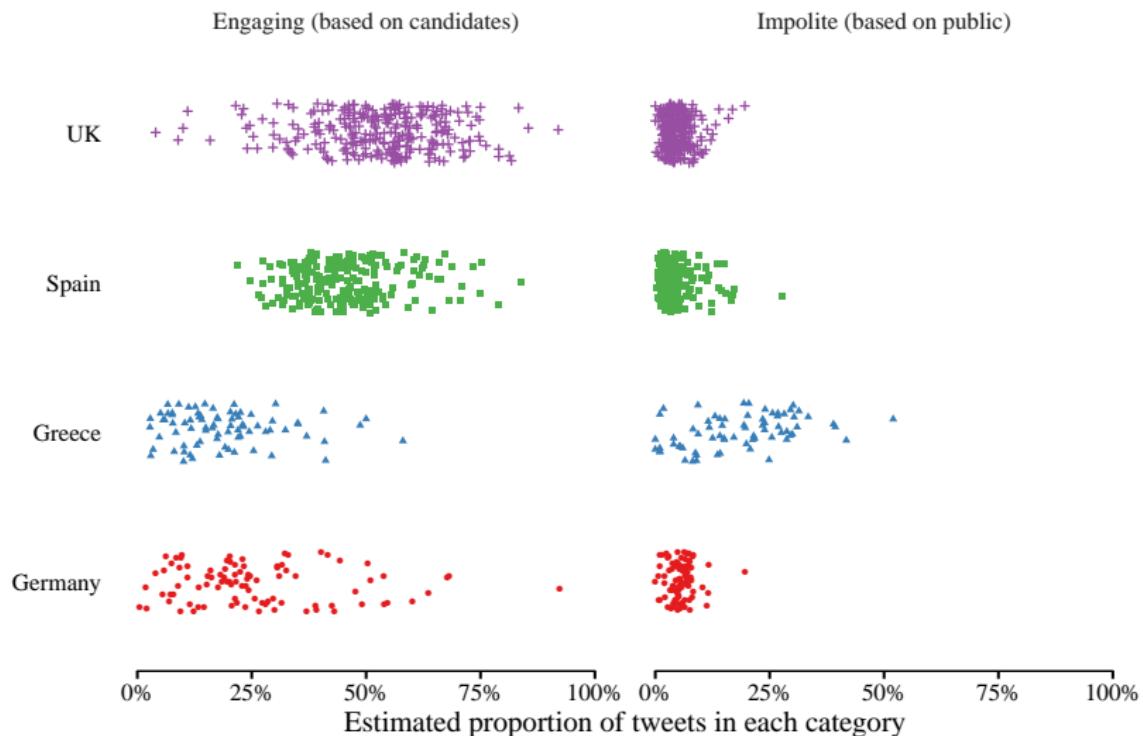
# Predictive validity

Citizens are more likely to respond to candidates when they adopt an engaging style



# Results: H1

Proportion of engaging tweets sent and impolite tweets received,  
by candidate and country



## Results: H2

Is engaging style positively related to impolite responses?

Three levels of analysis:

1. **Across candidates:** candidates who send more engaging tweets receive more impolite responses.
2. **Within candidates, over time:** the number of impolite responses increases during the campaign for candidates who send more engaging tweets
3. **Across tweets:** tweets that are classified as engaging tend to receive more impolite responses