Quantitative text analysis: Machine Learning for Text

Blake Miller

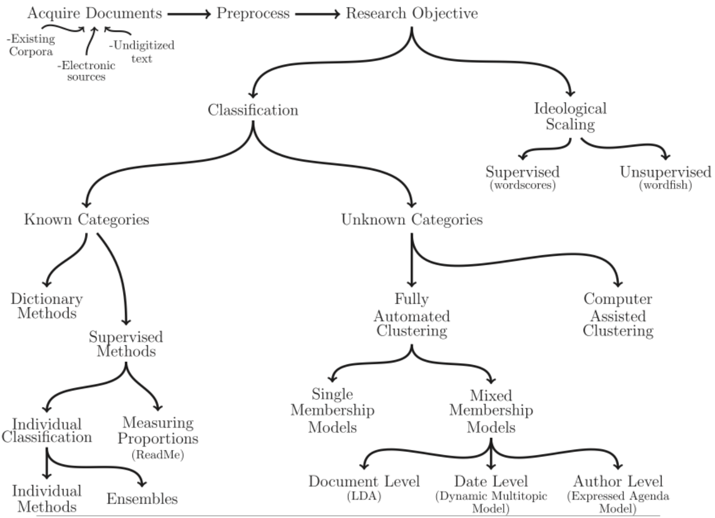
MY 459: Quantitative Text Analysis

February 10, 2020

Course website: 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
- Types of classifiers:
  - Naive Bayes
  - Regularized regression
  - Support Vector Machines (SVMs)
  - Ensemble classifiers

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, K-nearest neighbors, BART, ensemble methods...
- Approach to validate classifier: cross-validation
- Performance metric to choose best classifier and avoid overfitting: confusion matrix, accuracy, precision, recall...

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* require a training set that exemplify contrasting classes, identified by the researcher
  - *Unsupervised methods* scale documents based on patterns of similarity from the term-document matrix, without requiring a training step
- 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

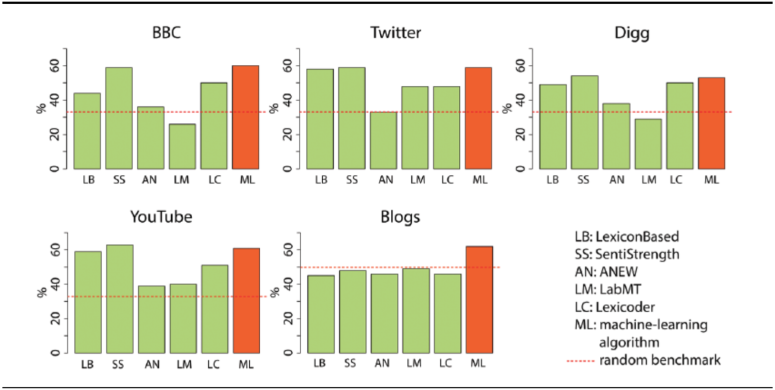
- General examples:
  - Supervised: Naive Bayes, regularized regression, Support Vector Machines (SVM)
  - Unsupervised: topic models, IRT models, correspondence analysis, factor analytic approaches
- Social science applications
  - Supervised: Wordscores (LBG 2003); SVMs (Yu, Kaufman and Diermeier 2008); 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

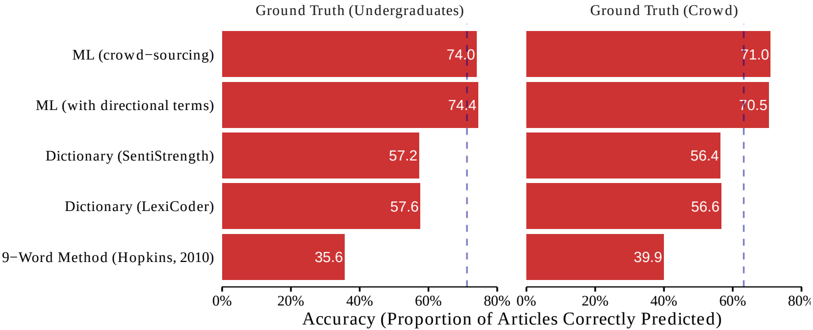
Lexicons' Accuracy in Document Classification
Compared to Machine-Learning Approach



Source: González-Bailón and Paltoglou (2015)

Dictionaries vs supervised learning

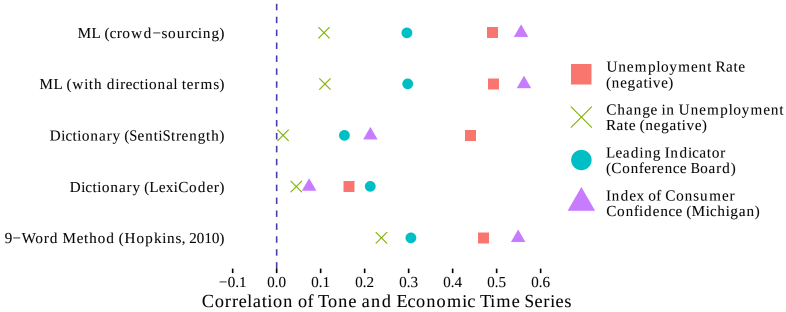
Application: sentiment analysis of NYTimes articles



Source: Barberá et al (2017)

Dictionaries vs supervised learning

Application: sentiment analysis of NYTimes articles



Source: Barberá et al (2017)

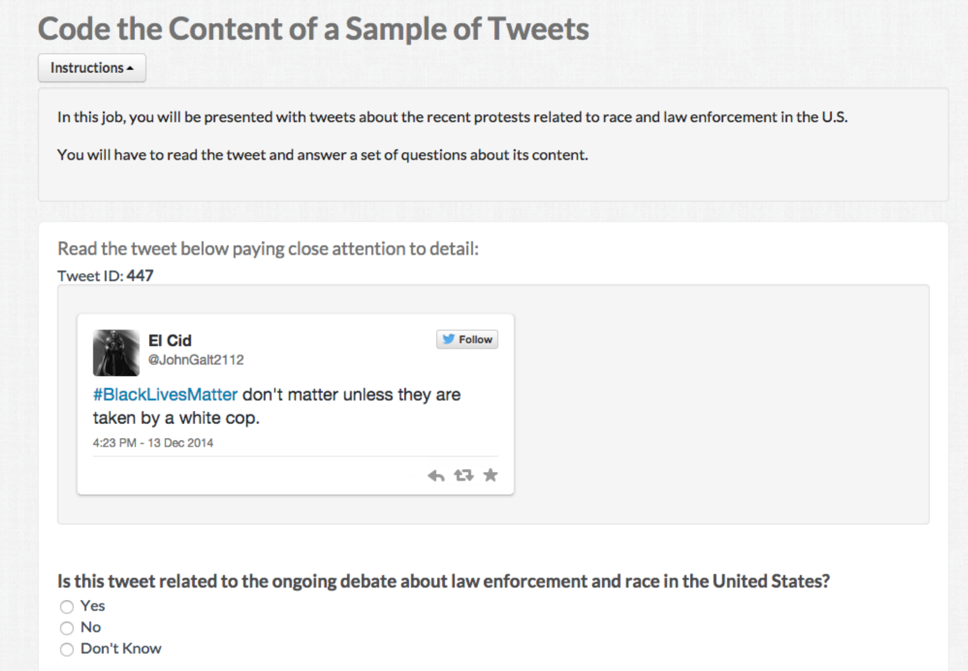
Outline

- Supervised learning overview
- Creating a labeled set and evaluating its reliability
- Classifier performance metrics
- Types of classifiers:
  - Naive Bayes
  - Regularized regression
  - Support Vector Machines (SVMs)
  - Ensemble classifiers

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

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

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

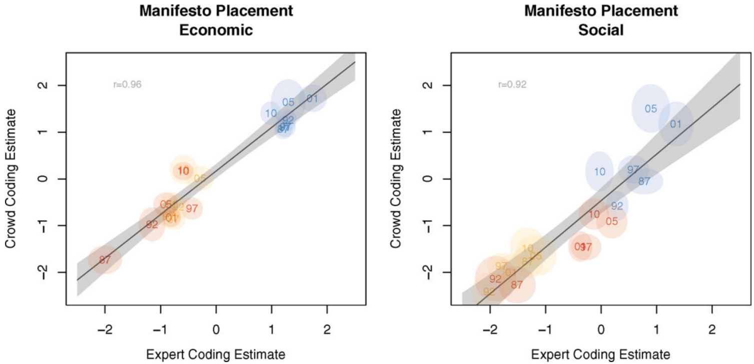
4:23 PM - 13 Dec 2014

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



Manifesto Placement
Economic

Manifesto Placement
Social

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

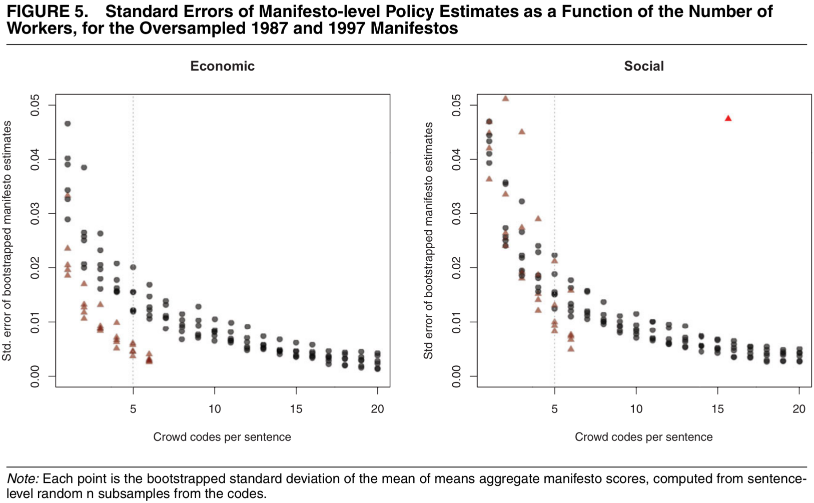


FIGURE 5. Standard Errors of Manifesto-level Policy Estimates as a Function of the Number of Workers, for the Oversampled 1987 and 1997 Manifestos

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-rate reliability, at three different standards:

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

Measures of agreement

- Percent agreement Very simple:

  (number of agreeing ratings) / (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 occured by chance*
  - Cohen's κ is most common
  - Krippendorf's α is a generalization of Cohen's κ
  - Both range from [0,1]

Reliability data matrixes

Example here used binary data (from Krippendorff)

Article:	1	2	3	4	5	6	7	8	9	10
Coder A	1	1	0	0	0	0	0	0	0	0
Coder 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 *dis*agreement: 4
- Expected *dis*agreement (by chance): 4.4211
- Krippendorff's $\alpha = 1 - \frac{D_o}{D_e} = 1 - \frac{4}{4.4211} = 0.095$
- Cohen's κ (nearly) identical

Outline

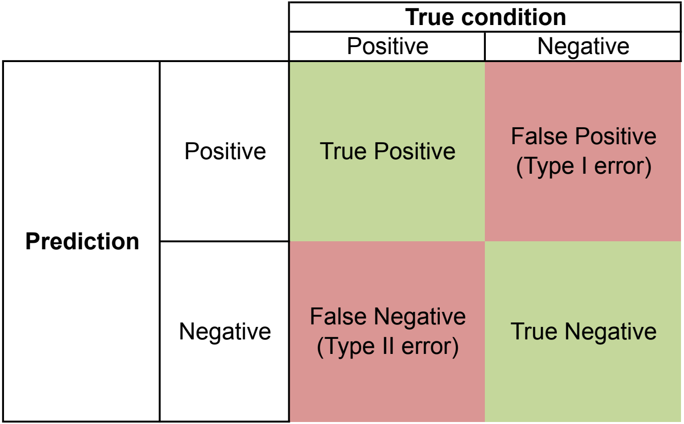
- Supervised learning overview
- Creating a labeled set and evaluating its reliability
- Classifier performance metrics
- Types of classifiers:
    - Naive Bayes
    - Regularized regression
    - Support Vector Machines (SVMs)
    - Ensemble classifiers

Basic principles of supervised learning

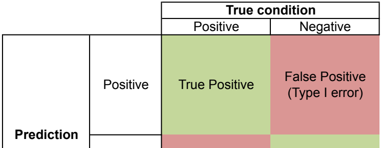
- Generalization: A classifier or a regression algorithm learns to correctly predict output from given inputs not only in previously seen samples but also in previously unseen samples
- Overfitting: A classifier or a regression algorithm learns to correctly predict output from given inputs in previously 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

- Confusion matrix:



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



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

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:

$$\begin{aligned} \text{Precision} &= (45/(45+15)) = 45/60 = 0.75 \\ \text{Recall} &= (45/(45+35)) = 45/80 = 0.56 \end{aligned}$$

Accuracy?

		True condition		
		Positive	Negative	
Prediction	Positive	🟩 45	🟥	60
	Negative	🟥	🟩	
		80		

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		
		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***

$$\begin{aligned} \text{Accuracy} &= (45+10)/105 && = 0.52 \\ \text{F1} &= 2*(0.75*0.56)/(0.75+0.56) && = 0.64 \end{aligned}$$

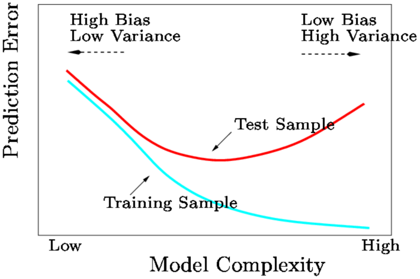
now assume 100 True Negatives:

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

$$\begin{aligned} \text{Accuracy} &= (45 + 100)/195 &&= 0.74 \\ \text{F1} &= 2 * (0.75 * 0.56)/(0.75 + 0.56) &&= 0.64 \end{aligned}$$

Measuring performance

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



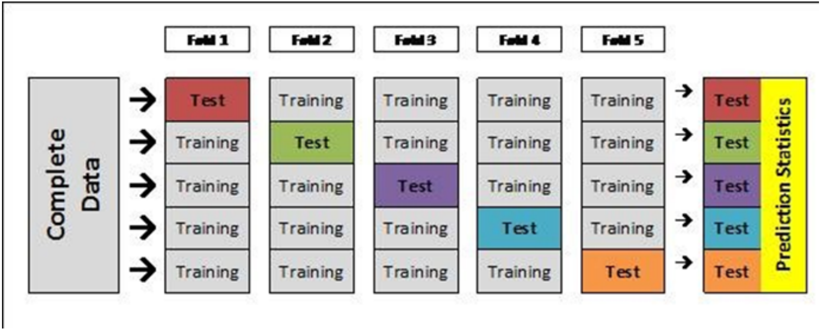
- Model is too complex, describes noise rather than signal (Bias-Variance trade-off)
- Focus on features that perform well in labeled data but may not generalize (e.g. "inflation" in 1980s)
- In-sample performance better than out-of-sample performance

- 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



Complete Data	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Prediction Statistics
	Test	Training	Training	Training	Training	Test
	Training	Test	Training	Training	Training	Test
	Training	Training	Test	Training	Training	Test
	Training	Training	Training	Test	Training	Test
	Training	Training	Training	Training	Test	Test

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)

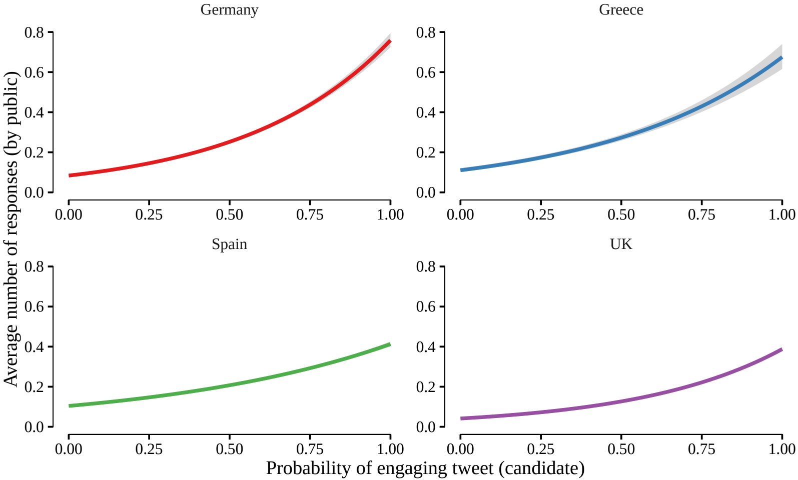
		UK	Spain	Greece	Germany
Communication Style	Accuracy	0.821	0.775	0.863	0.806
	Precision	0.837	0.795	0.838	0.818
	Recall	0.946	0.890	0.894	0.832
Polite vs. impolite	Accuracy	0.954	0.976	0.821	0.935
	Precision	0.955	0.977	0.849	0.938
	Recall	0.998	1.000	0.953	0.997
Morality and Democracy	Accuracy	0.895	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'v, you'v, @ 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

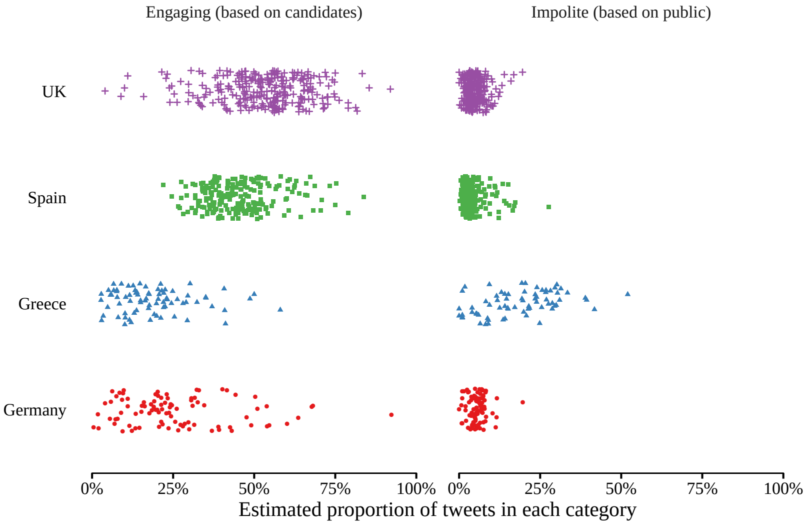
Predictive validity

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

Outline

- Supervised learning overview
- Creating a labeled set and evaluating its reliability
- Classifier performance metrics
- Types of classifiers:
  - Naive Bayes
  - Regularized regression
  - Support Vector Machines (SVMs)
  - Ensemble classifiers

Types of classifiers

General thoughts:

- Trade-off between accuracy and interpretability
- Parameters need to be cross-validated

Frequently used classifiers:

- Naive Bayes
- Regularized regression
- SVM
- Others: k-nearest neighbors, tree-based methods, etc.
- Ensemble methods

Outline

- Supervised learning overview
- Creating a labeled set and evaluating its reliability
- Classifier performance metrics
- Types of classifiers:
    - Naive Bayes
    - Regularized regression
    - Support Vector Machines (SVMs)
    - Ensemble classifiers

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

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

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 oberved 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, but 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:

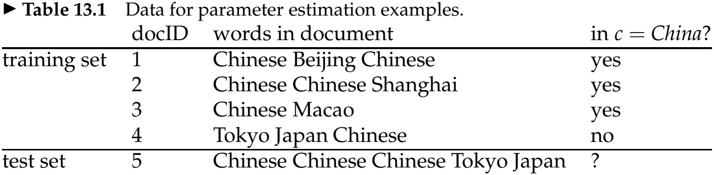
$$P(c|d) = P(c) \prod_j \frac{P(w_j|c)}{P(w_j)}$$

$$P(c|d) \propto P(c) \prod_j P(w_j|c)$$

- This is why we call it “naive”: because it (wrongly) assumes:
  - *conditional independence* of word counts
  - *positional independence* of word counts

Naive Bayes Classification Example

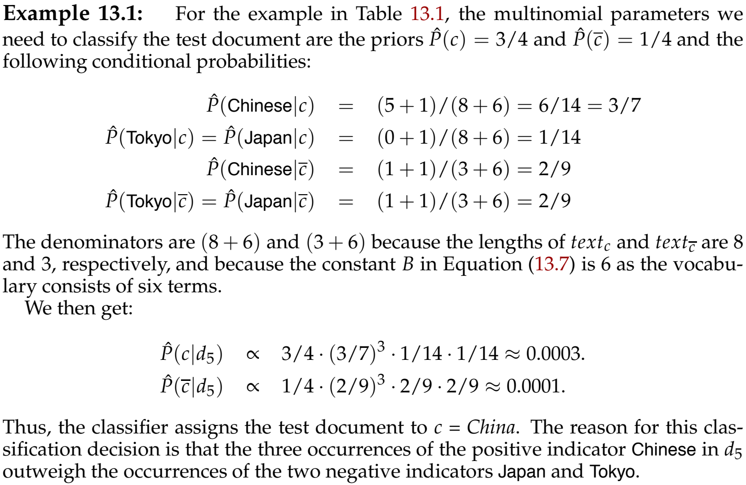
(From Manning, Raghavan and Schütze, *Introduction to Information Retrieval*)



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

	docID	words in document	in $c = China$?
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 $\hat{P}(\overline{c}) = 1/4$ and the following conditional probabilities:

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

The denominators are $(8+6)$ and $(3+6)$ because the lengths of $text_c$ and $text_{\overline{c}}$ 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:

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

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
- Types of classifiers:
    - Naive Bayes
    - Regularized regression
    - Support Vector Machines (SVMs)
    - Ensemble classifiers

Regularized regression

Assume we have:

- $i = 1, 2, \ldots, N$ documents
- Each document i is in class $y_i = 0$ or $y_i = 1$
- $j = 1, 2, \ldots, J$ 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 β_0, β_1, ..., β_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 λ is the **penalty parameter** (to be estimated)

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?)

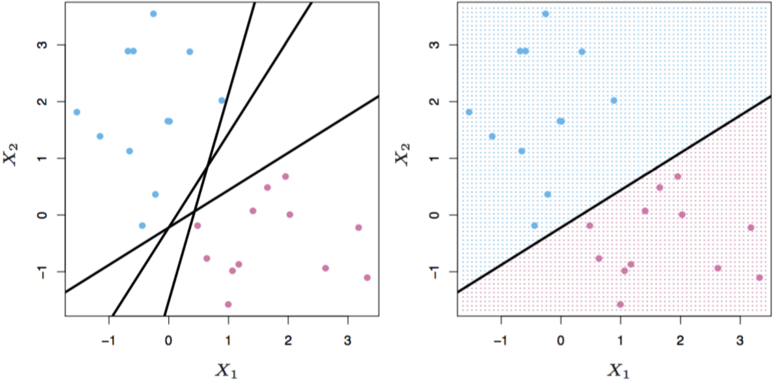
How to find best value of λ? Cross-validation.
Evaluation: regularized regression is easy to interpret, but often outperformed by more complex methods.

Outline

- Supervised learning overview
- Creating a labeled set and evaluating its reliability
- Classifier performance metrics
- Types of classifiers:
  - Naive Bayes
  - Regularized regression
  - Support Vector Machines (SVMs)
  - Ensemble classifiers

SVM

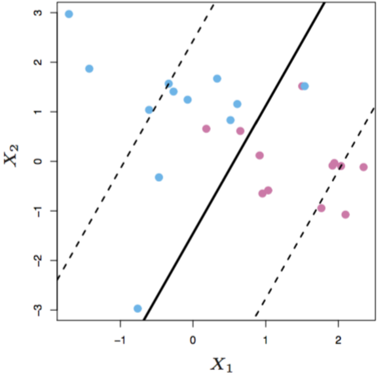
Intuition: finding classification boundary that best separates observations of different classes.

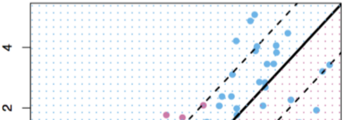


Harder to visualize in more than two dimensions (hyperplanes)

Support Vector Machines

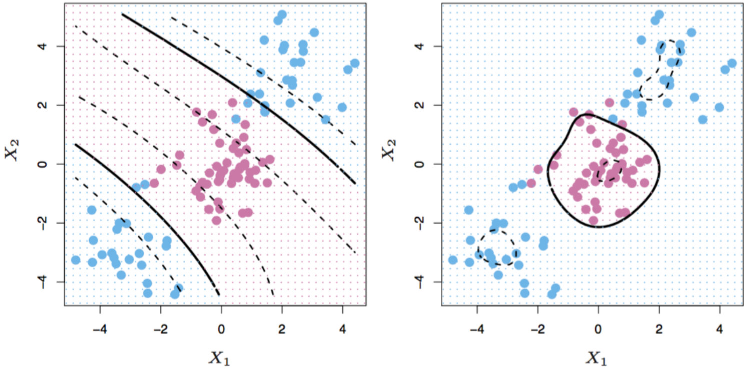
With no perfect separation, goal is to minimize distances to marginal points, conditioning on a tuning parameter C that indicates tolerance to errors (controls bias-variance trade-off)





SVM

In previous examples, vectors were linear; but we can try different kernels (polynomial, radial):



And of course we can have multiple vectors within same classifier.

Outline

- Supervised learning overview
- Creating a labeled set and evaluating its reliability
- Classifier performance metrics
- Types of classifiers:
  - Naive Bayes
  - Regularized regression
  - Support Vector Machines (SVMs)
  - Ensemble classifiers

Ensemble methods

Intuition:

- Fit multiple classifiers, different types
- Test how well they perform in test set
- For new observations, produce prediction aggregating predictions of individual classifiers
- How to aggregate predictions?
  - Pick best classifier
  - Average of predicted probabilities
  - Weighted average (weights proportional to classification error)
- Implement in `SuperLearner` package in R