Introduction to Automated Text Analysis

Pablo Barberá

School of International Relations
University of Southern California
`pablobarbera.com`

Lecture materials:
bit.ly/POIR599

Today

1. Solutions for last week's challenge
2. Reminder: project summary due next Monday
   - Two-page detailed summary of project
   - Submit via Blackboard
   - Email to your peer (see next slide)
   - Feedback: 2-3 paragraphs with your reaction
   - Feedback due by October 9th
3. Other announcements:
   - No class on November 21st
   - Office hours back at regular time tomorrow
4. Supervised learning
5. Unicode issues

Supervised Machine Learning.

Applications to text classification

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):
  - SVM, Naive Bayes, regularized regression, BART, ensemble methods...
- Approach to validate classifier: cross-validation
- Performance metric to choose best classifier and avoid overfitting: confusion matrix, AUC, accuracy, precision, recall...

Performance metrics

Confusion matrix:

	Actual label	
Classification (algorithm)	Negative	Positive
Negative	True negative	False negative
Positive	False positive	True positive

$$\text{Accuracy} = \frac{\text{TrueNeg} + \text{TruePos}}{\text{TrueNeg} + \text{TruePos} + \text{FalseNeg} + \text{FalsePos}}$$

$$\text{Precision}_{\text{positive}} = \frac{\text{TruePos}}{\text{TruePos} + \text{FalsePos}}$$

$$\text{Recall}_{\text{positive}} = \frac{\text{TruePos}}{\text{TruePos} + \text{FalseNeg}}$$

Performance metrics: an example

Confusion matrix:

	Actual label	
Classification (algorithm)	Negative	Positive
Negative	800	100
Positive	50	50

$$\text{Accuracy} = \frac{800 + 50}{700 + 50 + 100 + 50} = 0.85$$

$$\text{Precision}_{\text{positive}} = \frac{50}{50 + 50} = 0.50$$

$$\text{Recall}_{\text{positive}} = \frac{50}{50 + 100} = 0.33$$

Performance metrics: an example

Confusion matrix:

	Actual label	
Classification (algorithm)	Negative	Positive
Negative	800	100
Positive	50	50

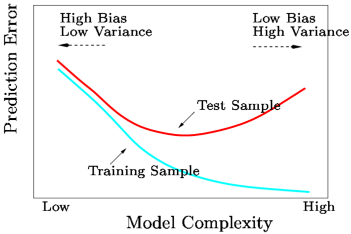
$$\text{Accuracy} = \frac{800 + 50}{700 + 50 + 100 + 50} = 0.85$$

$$\text{Precision}_{\text{positive}} = \frac{50}{50 + 50} = 0.50$$

$$\text{Recall}_{\text{positive}} = \frac{50}{50 + 100} = 0.33$$

Measuring performance

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



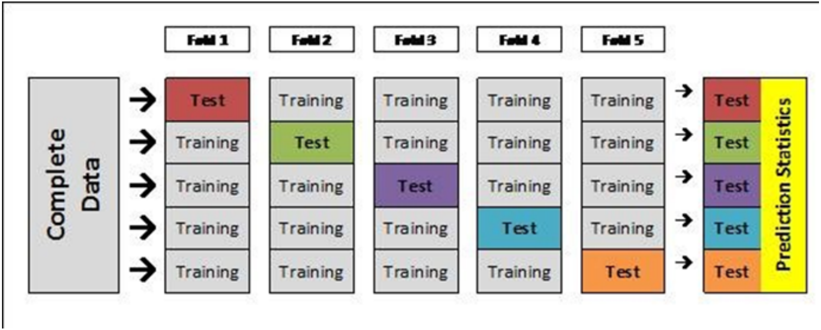
- Model is too complex, describes noise rather than signal (Bias-Variance trade-off)
- Focus on features that perform well in labeled data but may not generalize (e.g. "inflation" in 1980s)
- In-sample performance better than out-of-sample performance

- Solutions?
  - Split dataset into training and test set
  - Training dataset, random sample of entire dataset
  - 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.



Types of classifiers

General thoughts:

- It's just like regression!
- Trade-off between accuracy and interpretability
- Parameters need to be cross-validated

Frequently used classifiers:

- Regularized regression
- SVM
- Tree-based methods
- Ensemble methods

Regularized regression

Suppose we have N documents, with each document i having label $y_i \in \{-1, 1\} \rightsquigarrow \{\text{liberal, conservative}\}$
We represent each document i is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iJ})$.

$$
\begin{aligned}
f(\beta, \boldsymbol{X}, \boldsymbol{Y}) &= \sum_{i=1}^{N} \left(y_i - \beta' \boldsymbol{x}_i\right)^2 \\
\widehat{\beta} &= \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta' \boldsymbol{x}_i\right)^2 \right\} \\
&= \left(\boldsymbol{X}' \boldsymbol{X}\right)^{-1} \boldsymbol{X}' \boldsymbol{Y}
\end{aligned}
$$

Problem:

- J will likely be large (perhaps $J > N$)
- There many correlated variables

Source: Grimmer, 2014, “Text as Data” course week 15

Regularized regression

Penalty for model complexity

$$f(\beta, \boldsymbol{X}, \boldsymbol{Y}) \;=\; \sum_{i=1}^{N}\left(y_i - \;\beta_0 + \sum_{j=1}^{J}\beta_j x_{ij}\right)^2 + \underbrace{\lambda\sum_{j=1}^{J}\beta_j^2}_{\text{Penalty}}$$

where:

- $\beta_0 \rightsquigarrow$ intercept
- $\lambda \rightsquigarrow$ penalty parameter

Source: Grimmer, 2014, “Text as Data” course week 15

Regularized regression

Why the penalty (shrinkage)?

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

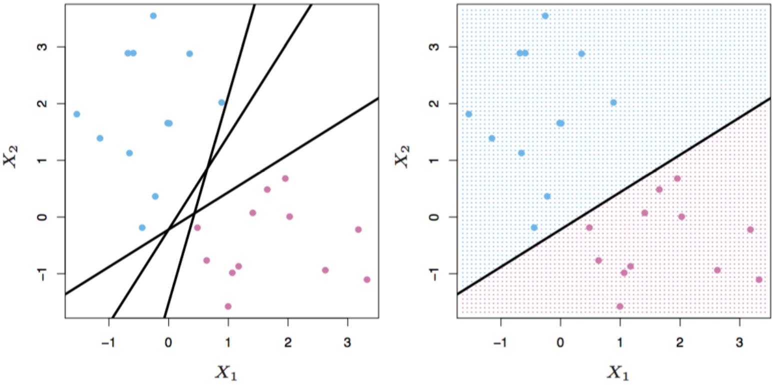
The penalty can take different forms:

- 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.

SVM

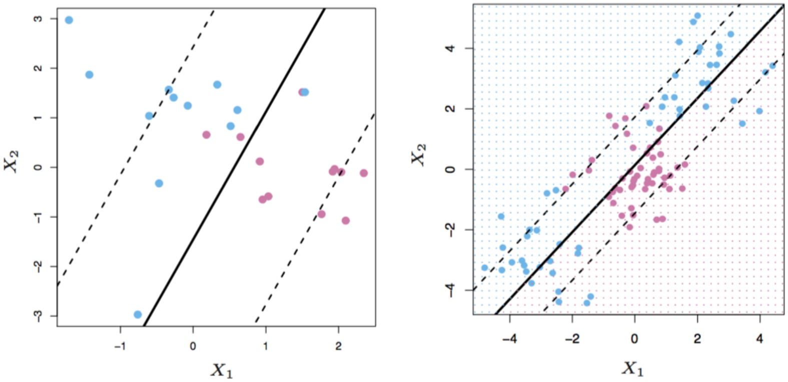
Intuition: finding best line that separates observations of different classes.



Harder to visualize in more than two dimensions (hyperplanes)

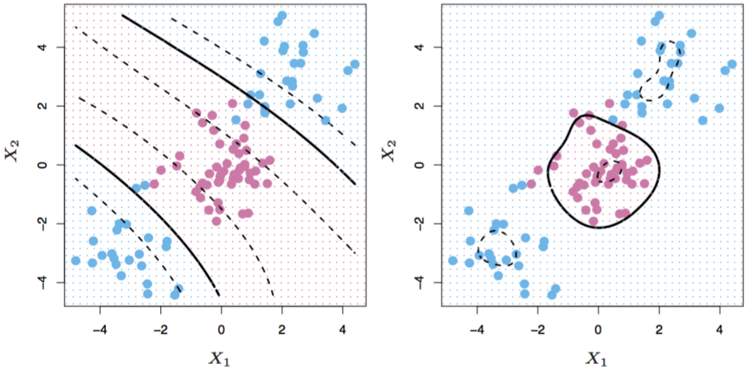
Support Vector Machines

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



SVM

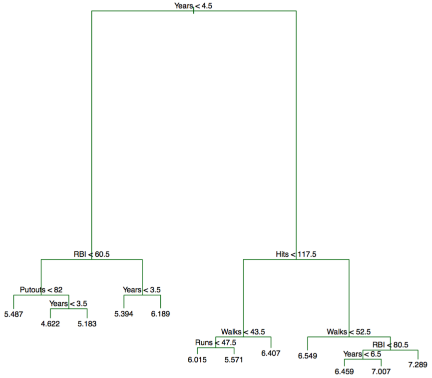
In previous examples, vectors were linear; but we can try different kernels (polynomial, radial):

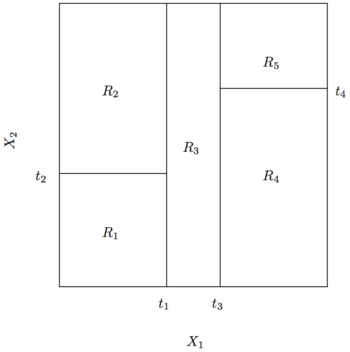


And of course we can have multiple vectors within same classifier.

Tree-based methods

Intuition: partition up dataset based on values of features





Different models answer questions differently:

- Where to split? And along what features?
- What should be the predicted value for each branch?

Ensemble methods

Ensemble methods

Process:

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