SOSC 4300/5500: Text Analysis; Supervised Machine Learning

Han Zhang

Oct 6, 2020

Outline

Logistics

- Please submit as a group for Assignment 1
-

Review

- Goal: classify documents into pre existing categories.
  - e.g. sentiment of tweets, ideological position of politicians based on manifestos
- We have seen how dictionary method works to classify documents into categories according to dictionaries
- But dictionary methods have many shortcomings
  - Constructing the set of matches in the dictionary is mostly a matter for human judgment
  - And it's quite often that words have multiple meanings

Supervised Machine Learning

- Supervised methods require a training that exemplify contrasting classes, coded by the researcher
- Formally, we have a training set of (X_i, Y_i), where each X is a document, and each Y is the category/outcome of the document X
- We train (fit/learn) an model f that maps X to Y: $Y = f(X)$
  - Hence machine learning
- With the trained model, the goal is to predict the outcomes of documents in the test set, whose categories are unknown

Supervised Machine Learning vs Dictionary Method

- Dictionary Method
  - Advantage: not corpus-specific, cost to apply to a new corpus is trivial
  - Disadvantage: not corpus-specific if you take an off-the-shelf dictionary, so performance on a new corpus is unknown (domain shift)
- Supervised learning
  - Advantage: corpus-specific
  - Disadvantage: You must already know the expected outcomes (e.g., what categories are allowed)

Supervised Machine Learning vs Dictionary Method

- Supervised machine learning can be conceptualized as a generalization of dictionary methods
  - Think about document-term matrix
  - Dictionary methods basically only keeps the columns whose words are in the dictionary;
  - Supervised methods keeps all words, and learn the weights of each column from data
    - Irrelevant words will then be assigned with lower weights
- Theoretically, supervised machine learning will outperform dictionary methods in classification tasks, as long as training set is large enough

docs	words made	because	had	into	get	some	through	next	where	many	irish
t06_kenny_fg	12	11	5	4	8	4	3	4	5	7	10
t05_cowen_ff	9	4	8	5	5	5	14	13	4	9	8
t14_ocaolain_sf	3	3	3	4	7	3	7	2	3	5	6
t01_lenihan_ff	12	1	5	4	2	11	9	16	14	6	9
t11_gormley_green	0	0	0	3	0	2	0	3	1	1	2
t04_morgan_sf	11	8	7	15	8	19	6	5	3	6	6
t12_ryan_green	2	2	3	7	0	3	0	1	6	0	0
t10_quinn_lab	1	4	4	2	8	4	1	0	1	2	0
t07_odonnell_fg	5	4	2	1	5	0	1	1	0	3	0
t09_higgins_lab	2	2	5	4	0	1	0	0	2	0	0
t03_burton_lab	4	8	12	10	5	5	4	5	8	15	8
t13_cuffe_green	1	2	0	0	11	0	16	3	0	3	1
t08_gilmore_lab	4	8	7	4	3	6	4	5	1	2	11
t02_bruton_fg	1	10	6	4	4	3	0	6	16	5	3

Steps in supervised methods

Supervised	Dictionary
Collect corpus and preprocess	the same
Collect training data	Collect/construct dictionaries
Fit algorithms	Apply dictionary on corpus
Validation	Validation

How do we obtain a labeled training set?

- Usually we already have (texts)
- How can we get the category for these texts?
- External sources of annotation
- Jake M. Hofman, Amit Sharma, and Duncan J. Watts, *Prediction and explanation in social systems*, Science **355** (2017), no. 6324, 486–488
- Y is cascade size: number of total retweets (including retweets of retweets, and so on so fort)
- X is text of tweets, user info, past number of retweets.
- In other words, Y is automatically obtained by some external process
- Other examples?
- Cheap labels are usually noisy; do not always contain what you want

Expert annotation

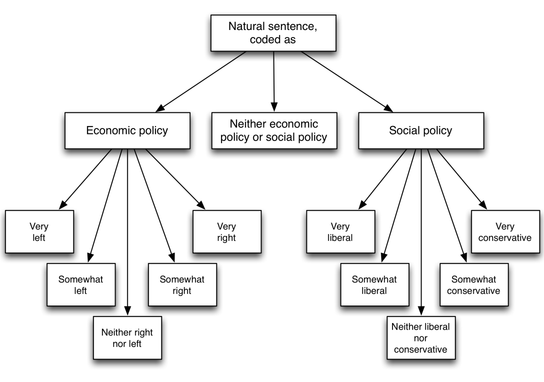
- If you cannot find existing labeled training data that fits your need
- Expert annotation
- E.g., the Comparative Manifesto Project
  - Texts: parties' election manifestos in major electoral democracies
  - Outcomes: a bunch of variables related to the party's policy preferences reflected in the texts.
  - 4,000 manifestos from nearly 1,000 parties in 50 countries and then organized political scientists to systematically code them. Each sentence in each manifesto was coded by an expert using a 56-category scheme
  - `https://manifestoproject.wzb.eu/down/tutorials/primer.html`
- In most academic projects, PG/UG students with some training do the coding

Crowd-sourced coding

- Crowd-sourced coding:
  - Wisdom of crowds: aggregated judgments of non-experts converge to judgments of experts at much lower cost
- E.g., crowd-sourced coding of the Comparative Manifesto Project
- Kenneth Benoit, Drew Conway, Benjamin E. Lauderdale, Michael Laver, and Slava Mikhaylov, *Crowd-sourced Text Analysis: Reproducible and Agile Production of Political Data*, American Political Science Review **110** (2016), no. 2, 278–295
  - crowd-source workers were asked to classify each sentence as referring to economic policy (left or right), to social policy (liberal or conservative), or to neither
  - Key here: simplify the burden for coder! 56 categories are too much for non-experts.

Crowd-sourced coding

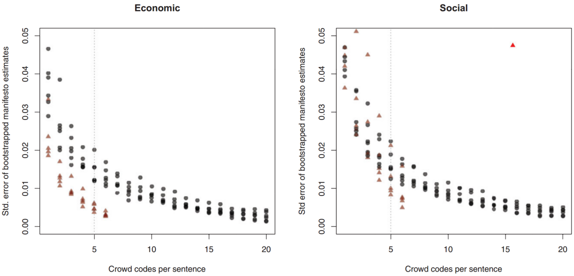
:PROPERTIES: :CUSTOM_ID:
h:6e0924f1-7b65-49d0-b350-38bdba466572



Crowd-sourced coding

- With more coders on the same document, error of coding decreases

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.

Crowd-sourced coding

- The crowd-source coding produce high-quality results, on par with expert coding
- And it is quick
- E.g., Benoit et al. want to code a new variable related to immigrants
  - "Within 5 hours of launching their project, the results were in. They had collected more than 22,000 responses at a total cost of $360", based on around 51 coders

How many labeled document is enough?

- Depending on problems.
- (always try some dictionary methods first!)
- First collect several hundreds or a thousand, if your Y is binary
- Then start to fit some models and see performances
- And see if you need to code more

Various algorithms you will commonly used

- We have introduced how do you transform text data into a matrix X in the last lecture
- And we have labels (Y)
- Then we can use the algorithms that you have used for Assignment 1 to make prediction for texts
  - Linear/logistic regression
  - LASSO and Elastic Net: linear regression
  - Tree and Forests
  - SVM

LASSO and Elastic Net

- We have p variables
- Linear regression:

$$\hat{\beta}_{OLS} = argmin_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 \tag{1}$$

- Lasso estimator

$$\hat{\beta}_{LASSO} = argmin_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| \tag{2}$$

- ElasticNet

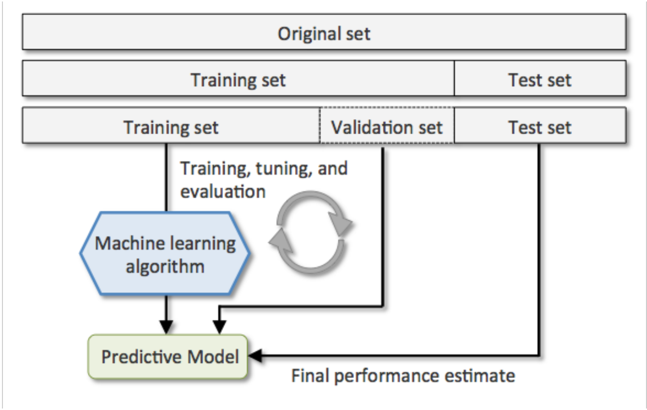
$$\hat{\beta}_{Ridge} = argmin_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \tag{3}$$

LASSO and Elastic Net

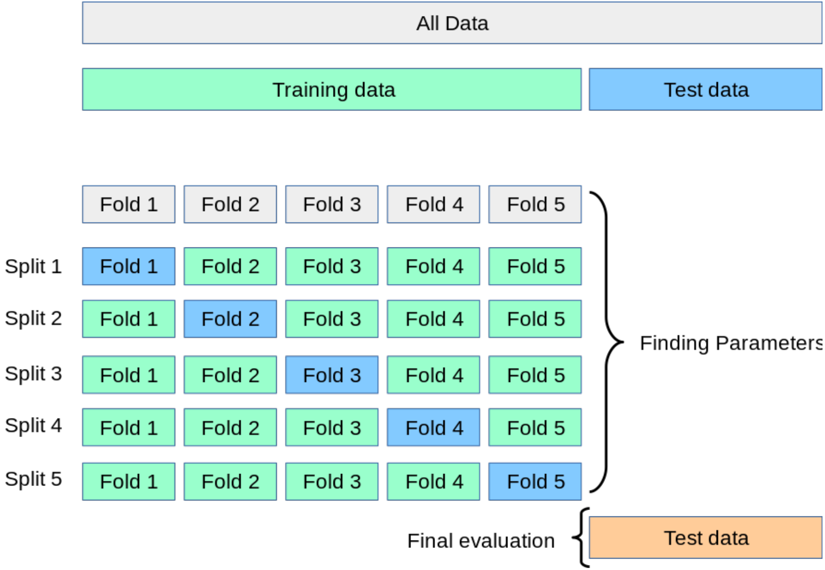
- Forcing coefficient estimates to be small is called regularization, or adding penalty
- Larger λ_1 forces more coefficients to be 0
- Larger λ_2 forces more coefficients to be small (but not 0)
- How do we choose the value of λ ?
- These are often called hyperparameters;
  - We do not care about their values
  - But we need to set them in order to run our algorithms

Train/validation/test split to select

- We use validation data to select the values of hyperparameters

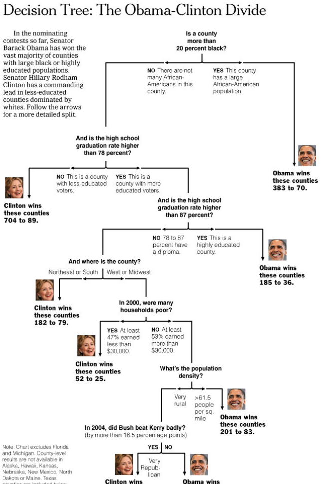


Cross validation

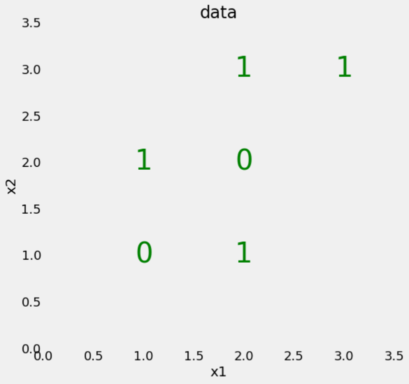


Decision Tree Example

https://archive.nytimes.com/www.nytimes.com/imagepages/2008/04/16/us/20080416_OBAMA_GRAPHIC.html?emc=polb1&nl=pol



Growing a Tree



- The above data cannot easily be separated by drawing a straight line (i.e., simplest linear regression)
- Let us draw a tree by ourselves to distinguish $Y = 0$ vs. $Y = 1$, based on X_1 and X_2
    - We want a binary tree: split into two branches

Some details

1. There are usually multiple ways to grow a tree
   - How do we pick the order we draw branches?
2. We usually work with binary tree. Otherwise:
   - For continuous X, we can split it in many ways
   - For categorical X, if the number of levels is large, we can still have a very wide tree
3. What if we there are multiple outcomes on a same leaf?
   - For continuous outcomes, the prediction is the mean
   - For categorical outcomes, the prediction is the mode
4. No need to use all predictors: tree automatically does variable selection
5. One predictor can be used multiple times

Desision Tree Algorithms

- Decision Tree Algorithms help you to draw a tree from more complex data
- What are the steps we should take
- We have a list of features $X_1, \cdots X_m$
- Which feature X_j to choose first?
- The intuitive answer is that:
  - Choose X_j that best separates Y
    - i.e., the left branch is mostly one category and the right branch is mostly the other category
  - Thus predicts Y the best)

Some details

- The above procedure tells you how to grow the first branch
- For any internal nodes, the procedure is the same, expect that you only use observations that belong to this branch to
  - This is known as *greedy recursive binary splitting*
  - It is possible that a particular choice of X_j may not be the current best, but may turn out to be the best choice in the future (locally best vs. globally best)
  - But we do not consider the above possibility; just *greedily* choose the best partition and do not look back
- Intuitively, whenever we grow a new branch, we are adding more interactions, using regression analogy.

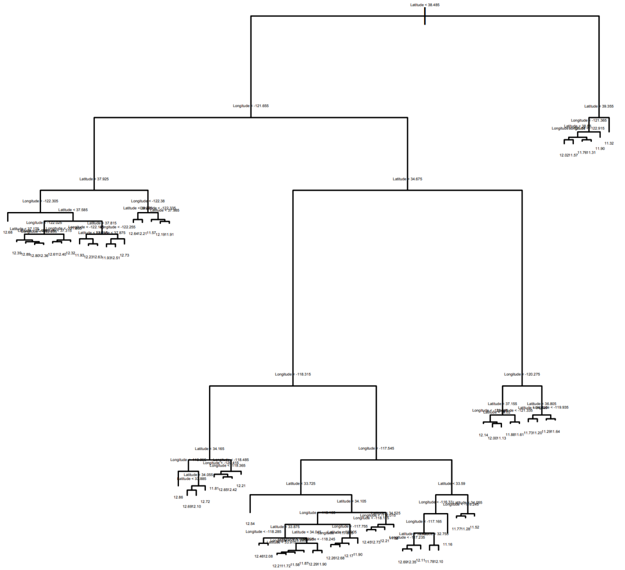
Decision Tree: Over-fitting and Regularization

- The tree grown using the above procedure can be quite complex
- We can always make a very complex tree by:
  - Try your best to make every single leaf contains only one Y
- In this way, we overfit the data
  - Complex trees fit the data nearly perfectly
  - But it does not generalize well to the test set
    - Each time you have a new observation, the tree may look entire different
- We need to regularize to make tree simpler
- In decision trees, people often call regularization as pruning

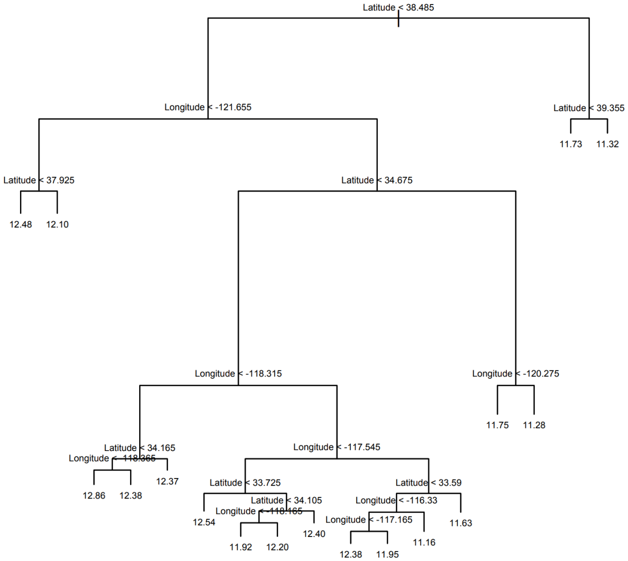
Pruning strategy

- Intuitively, we are pruning leaves that are too small
- Many different ways to make your tree simpler:
  - Force the tree to have no more than certain number of leaves
  - Force the tree to have no more than certain depth

Decision Tree: un-pruned



Decision Tree: pruned



Pros and Cons of Decision Trees

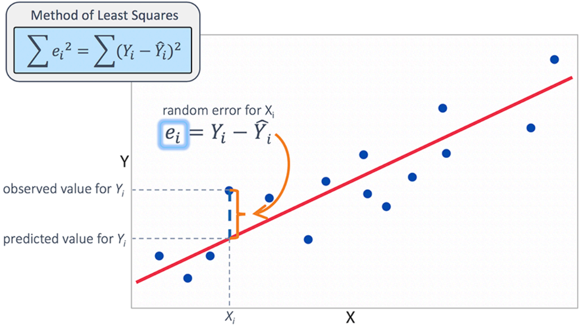
- Pros:
  - visualize
  - easy to recognize which feature is more important (those on top of a tree)
  - easy to understand
  - handles complex interactive data with ease
  - handles categorical variables easily: no need for dummies
- Cons:
  - If your data are not that complex, tree easily overfit
  - And it is slow

Random Forest

- Random Forests further extend the idea of bagging
- The key innovation of random forests:
- For each sample from the original training data, randomly select m variables (not using all p variables), and grow a tree;
  - A common choice: $m = \sqrt{p}$
- In other words, we just force $p - m$ predictors to be non-relevant each time
- Why? High-dimensional data!

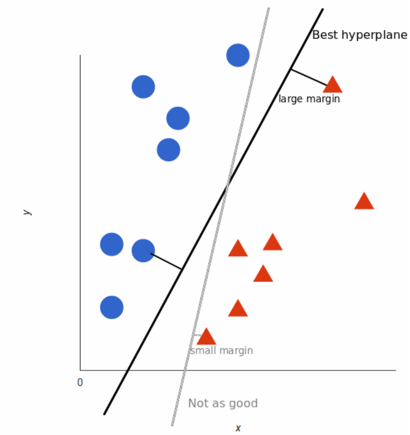
SVM: linear case

- Least square: minimize mean square error



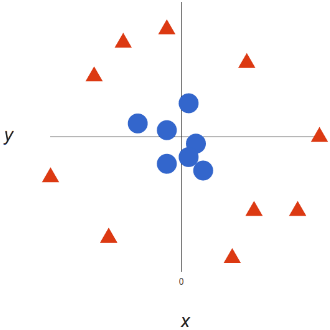
SVM: linear case

- SVM: maximize margin



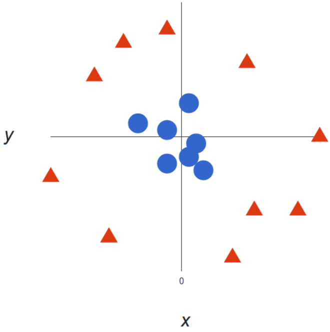
SVM: nonlinear case

- Some data are not linearly separable
- That is, it's mathematically impossible that you write a linear/logistic regression and use different interactions of X to perfectly classify Y



SVM: kernel tricks

- SVM performs kernel tricks
  - By projecting data to a higher-dimension space
  - And then the projected data becomes linearly separable
  - `https://towardsdatascience.com/mathematics-behind-svm-support-vector-machines-84742ddd`



SVM: practice

- First developed for binary classification; some extensions are made for multiple
- Has dominated the CS literature for a while (in the 90s and early 00s)
- Can be slower than LASSO
- Commonly used kernels
  - Radial basis function kernel (RBF): more explicitly project the data onto higher dimension, thus is more powerful
    - but slow
  - Linear/polynomial kernel: less powerful