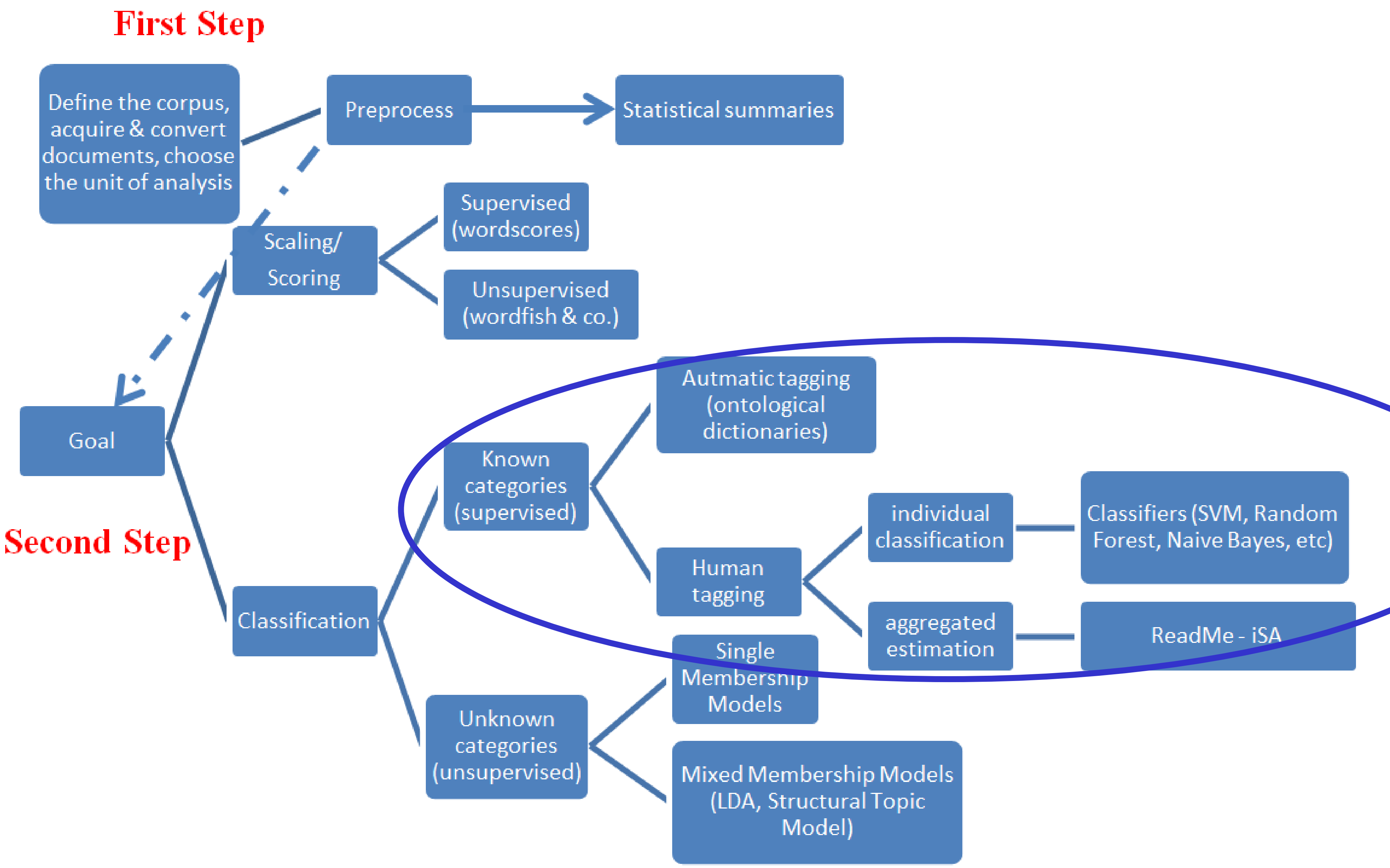
Big Data Analytics

Lecture 9 (part 2)

The importance of the training-set

Our Course Map


First Step
Define the corpus, acquire & convert documents, choose the unit of analysis
Preprocess
Statistical summaries
Scaling/ Scoring
Supervised (wordscores)
Unsupervised (wordfish & co.)
Goal
Second Step
Autmatic tagging (ontological dictionaries)
Known categories (supervised)
individual classification
Classifiers (SVM, Random Forest, Naive Bayes, etc)
Human tagging
aggregated estimation
ReadMe - iSA
Classification
Single Membership Models
Unknown categories (unsupervised)
Mixed Membership Models (LDA, Structural Topic Model)

Reference

- ✓ Grimmer, Justin, and Stewart, Brandon M. (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, 21(3): 267-297
- ✓ Curini, Luigi, and Robert Fahey (2020). Sentiment Analysis and Social Media. In Luigi Curini and Robert Franzese (eds.), *SAGE Handbook of Research Methods is Political Science & International Relations*, London, Sage, chapter 29
- ✓ Barberá, Pablo et al. (2020) Automated Text Classification of News Articles: A Practical Guide, *Political Analysis*, DOI: 10.1017/pan.2020.8

Supervised Learning vs. Dictionary methods

This approach to classification has **four major advantages** over dictionary methods:

First, it is **necessarily domain specific** and therefore avoids the problems of applying dictionaries outside of their intended area of use

Applying supervised learning methods **requires scholars to develop coding rules for the particular quantities of interest** under study when working on the training-set

This also forces scholars to develop coherent definitions of concepts for particular applications, which leads to clarity in what researchers are measuring and studying

Supervised Learning vs. Dictionary methods

Second, when using a dictionary you are relying **only** on the words (and on their relative weight: +1 or -1 for example) as defined in the dictionary

That is…the dictionary comes with a hard-wired set of parameter values for the importance of a predetermined set of features

In contrast, when using a supervised approach the relevant features of the text and their weights are estimated directly from the data (in the training set). The feature space is thus likely to be both larger and more comprehensive than that used in a dictionary

The end result is that much more information drives the subsequent classification of text

Supervised Learning vs. Dictionary methods

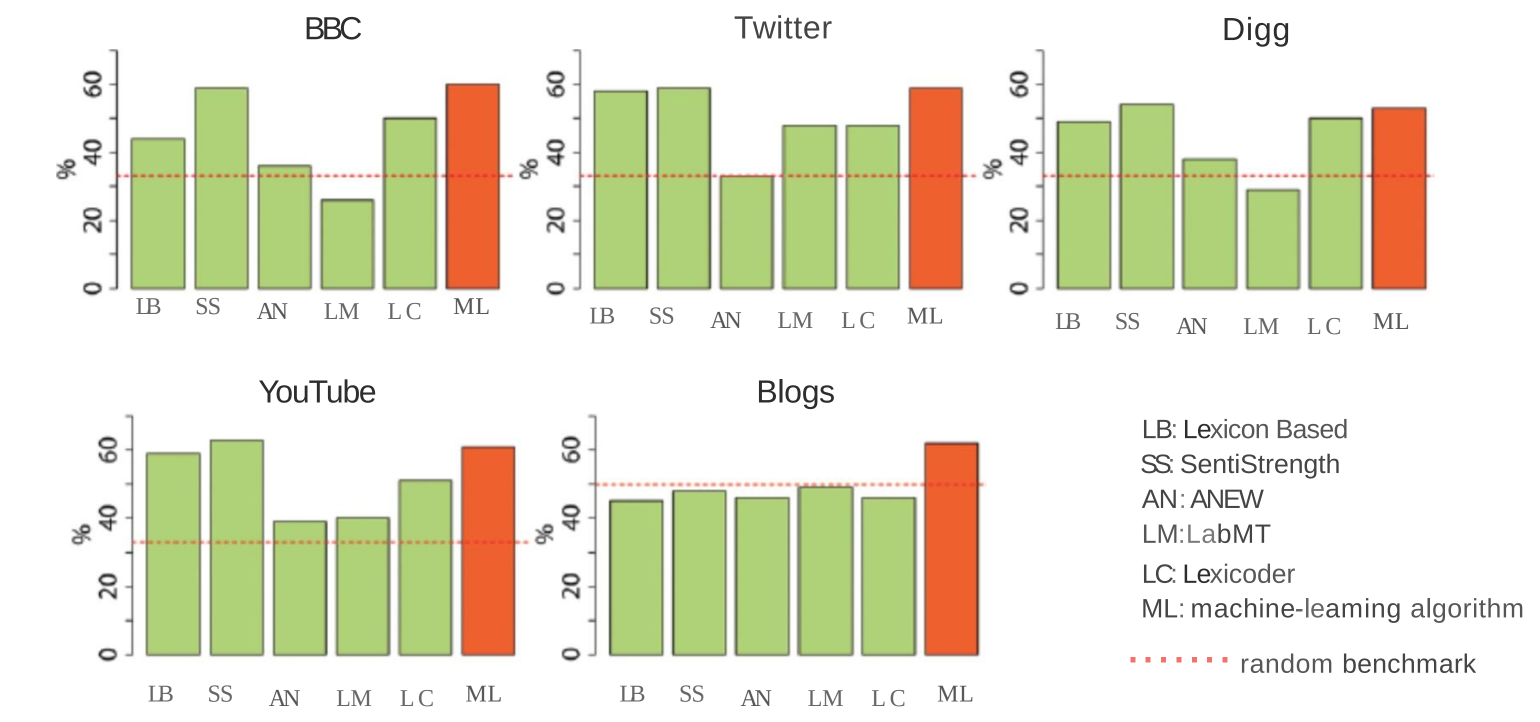
In particular, the supervised learning model will estimate parameter values optimized to minimize error of the classifier on the training dataset

Thus, they will necessarily outperform the dictionary on that **same sample**, as long as training sample is large enough

And indeed…

Supervised Learning vs. Dictionary methods

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



Source: Gonzalez-Bailn and Paltoglou (2015)

Supervised Learning vs. Dictionary methods

But which does better **out-of-sample**?

Here, too, since a supervised learning model is trained on a sample of the data, it is guaranteed to do better than a dictionary as long as it is trained on a **large enough random sample**

Indeed, as the sample converges to the population - or as the training dataset contains an ever increasing proportion of words encountered - a supervised learning model has to do better than a dictionary, as the estimated parameter values will converge to the true parameter values

Supervised Learning vs. Dictionary methods

Third, human involvement is crucial to understand the correct meaning of a text (double meaning sentences, specific jargons, neologisms, irony)

Fourth, supervised learning methods are much easier to validate, with clear statistics that summarize model performance (as we have discussed)

Supervised Learning vs. Dictionary methods

While a dictionary cannot compete with a classifier trained on a **representative and large enough training dataset**, in any given task dictionaries may however outperform a supervised learning model **if these conditions are not met**

Dictionaries bring rich prior information to the classification task: humans may produce a topic-specific dictionary that would require a large training dataset to outperform it

Supervised Learning vs. Dictionary methods

Summing up:

Dictionaries:

- ✓ Can be off the-shelf
- ✓ no creation of a training dataset required
- ✓ easy to apply to a given corpus
- ✓ built by humans who can bring domain expertise to bear

Supervised Learning vs. Dictionary methods

Summing up:

Supervised machine learning:

- ✓ optimized for current research question
- ✓ more comprehensive set of features used to classify text
- ✓ mathematically, ML necessarily outperforms dictionary methods given a large enough training dataset
- ✓ by construction, the analyst knows the performance of the classifier based on multiple measures of fit (i.e, how closely the labels generated correspond to human coding)

Supervised Learning vs. Dictionary methods

Summing up:

Supervised learning can be therefore conceptualized as a **generalization of dictionary methods**, where features associated with each categories (and their relative weight) are learned from the data **via human intervention**

Constructing a training set

For supervised problems, the researcher is aiming to classify documents into a set of known or assumed categories based upon rules or information that can be learned from the training set

This requires **labels** in the training set from which to infer categories in the test set

The most important step in applying a supervised learning algorithm is therefore constructing a **reliable training set**, because no statistical model can repair a poorly constructed training set!

If the training set is **poorly constructed**, the supervised algorithms will simply replicate such poorly construction!

Constructing a training set

(1) creating and executing a coding scheme:

Best practice is to **iteratively develop coding schemes**

Initially, a **concise codebook** is written to guide coders, who then apply the codebook to an initial set of documents

When using the codebook, particularly at first, coders are likely to **identify ambiguities** in the coding scheme or overlooked categories

Constructing a training set

(1) creating and executing a coding scheme:

This subsequently leads to a **revision of the codebook**, which then needs to be applied to a new set of documents to ensure that the ambiguities have been sufficiently addressed

Only after coders apply the coding scheme to documents without noticing ambiguities is a “final” scheme ready to be applied to the data set

Constructing a training set

(2) **sampling documents:**

Almost all (but not all…) classification methods implicitly assume that the **training set is a random sample** from the population of documents to be coded

This is because Supervised learning methods **use the relationship** between the features in the training set to classify the remaining documents in the test set (out-of-sample predictions)

Constructing a training set

(2) **sampling documents:**

This presents particular difficulty when…

…**all the data are not available at the time of coding**: either because it will be produced in the future or because it has yet to be digitized

Per-se, this could be particularly problematic in dealing with any ***semantic change***, which is the difference in the meaning of language between the labeled and unlabeled sets

Constructing a training set

For example, we can have **emergent discourse**, where new words and phrases, or the meanings of existing words and phrases, appear in the unlabeled set but not the labeled set, and **vanishing discourse**, where the words, phrases, and their meanings exist in the labeled set but not the unlabeled set

Constructing a training set

Russian election hacking after 2016 US Presidential elections is an example of emergent discourse, language which did not exist a few years ago, whereas *Russian Communism* is an example of vanishing discourse, with language that has largely vanished from ongoing conversations over time

How to face this risk?

Keep updating the training-set (if your test-set is still to come...)!

Constructing a training set

(2) **sampling documents:**

Moreover, Supervised methods need **enough information** to learn the relationship **between words and documents in each category of a coding scheme**

Hopkins and King (2010) offer **five hundred** as a rule of thumb with one hundred documents probably being enough

Constructing a training set

(2) **sampling documents:**

Still the number necessary will depend upon **the specific application of interest**. For example, as the number of categories in a coding scheme increases, the number of documents needed in the training set also increases

Moreover, if a category does not occur, or occurs extremely rarely, in the training set, there is insufficient opportunity to “learn” about this category and its properties, which will in turn interfere with the process of classifying test-set documents into this category correctly

Constructing a training set

(2) **sampling documents:**

When attempting to detect **small changes or rare categories**, *therefore*, increasing the probability that they are observed in the training set often means increasing the size of the training set relative to the test set (remember the curse of highly imbalanced training-set!)

Constructing a training set

(2) **sampling documents:**

You also have to consider the risk of a ***lack of textual discrimination***. This happens where the language used in documents falling in the different pre-defined categories is not clearly distinguishable

Lack of textual discrimination among categories can occur because of heterogeneity in how authors express category-related information or a divergence between how authors of the documents express this information and how the analyst conceptualizes the categories

Also this factor affects the appropriate size of the training-set

Constructing a training set

(3) **checking human-tagging reliability:**

Manually generating the initial set of labels can prove arduous and time-consuming, but is also fraught with concerns about consistency and accuracy

That is, while labeling training data often requires the use of human coders to sort texts into desired categories, human coding lacks consistency and reliability both within and across individuals, above and beyond the time and expense required to complete the task

Therefore always run an **inter-coder reliability text**!!!

Constructing a training set

(3) **checking human-tagging reliability:**

What is **inter-coder (or inter-rater) reliability**?

Intercoder reliability is the extent to which 2 different researchers agree on how to code the same content

It's often used in content analysis when one goal of the research is for the analysis to aim for consistency and validity

Intercoder reliability ensures that when you have multiple researchers coding a set of data, that they come to the same conclusions

Constructing a training set

One common statistics used is Cohen's kappa coefficient (κ)

k is a more robust measure than simple percent agreement calculation, as it takes into account the possibility of the agreement occurring by chance

How? K is estimated as $(p_o-p_e)/(1- p_e)$

where p_o is the relative observed agreement among raters (identical to accuracy), and p_e is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly seeing each category

If the raters are in complete agreement then k=1. If there is no agreement among the raters other than what would be expected by chance (as given by p_e), k=0. It is possible for the statistic to be negative, which implies that the agreement is worse than random

Usually a reasonable value for k is larger than .6 (but larger than .8 would be far better)

Constructing a training set

Let’s see an example, with 2 coders, 2 categories, and 25 texts to code for each coders

	Coder B	
Coder A	Positive	Negative
Positive	20	5
Negative	10	15

Observed proportionate agreement (p_o): (20+15)/50=0.7

Constructing a training set

	Coder B	
Coder A	Positive	Negative
Positive	20	5
Negative	10	15

And the probability of a random agreement (p_e)?

- ✓ Coder A said “Positive" to 25 texts and “Negative" to 25 texts. Thus reader A said “Positive" 50% of the time.
- ✓ Reader B said “Positive" to 30 texts and "Negative" to 20 texts. Thus reader B said “Positive" 60% of the time

So the expected probability that both would say “Positive” at random is: 0.5*0.6=0.3

Similarly, the expected probability that both would say “Negative” at random is: 0.5*0.4=0.2

Overall random agreement probability is the probability that they agreed on either Positive or Negative, i.e. (p_e)=0.3+0.2=0.5

Constructing a training set

	Coder B	
Coder A	Positive	Negative
Positive	20	5
Negative	10	15

Applying the formula for Cohen's Kappa we get:

✓ $k=(p_o-p_e)/(1-p_e)=(0.7-0.5)/(1-0.5)=0.4$

Constructing a training set

Given the likely existence of a budget constraint sometimes we will need to make a **choice between more coders per object and more texts coded**. What is better?

The literature shows that while increasing the number of coders for each document can improve the accuracy of the classifier, the informational gains from increasing the number of documents coded are greater than from increasing the number of codings of a given document

Constructing a training set

This does not obviate the need to have multiple coders code at least a subset of documents, namely to determine coder quality and to select the best set of coders to use for the task at hand

But once the better coders are identified, the optimal strategy is to proceed with one coder per document

R pakcages to install

install.packages("irr", repos='http://cran.us.r-project.org')