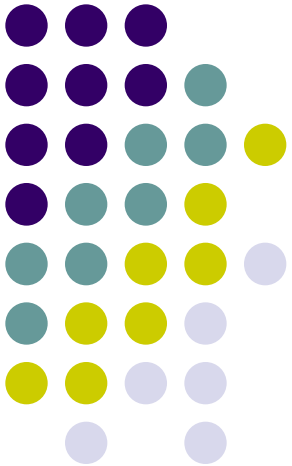
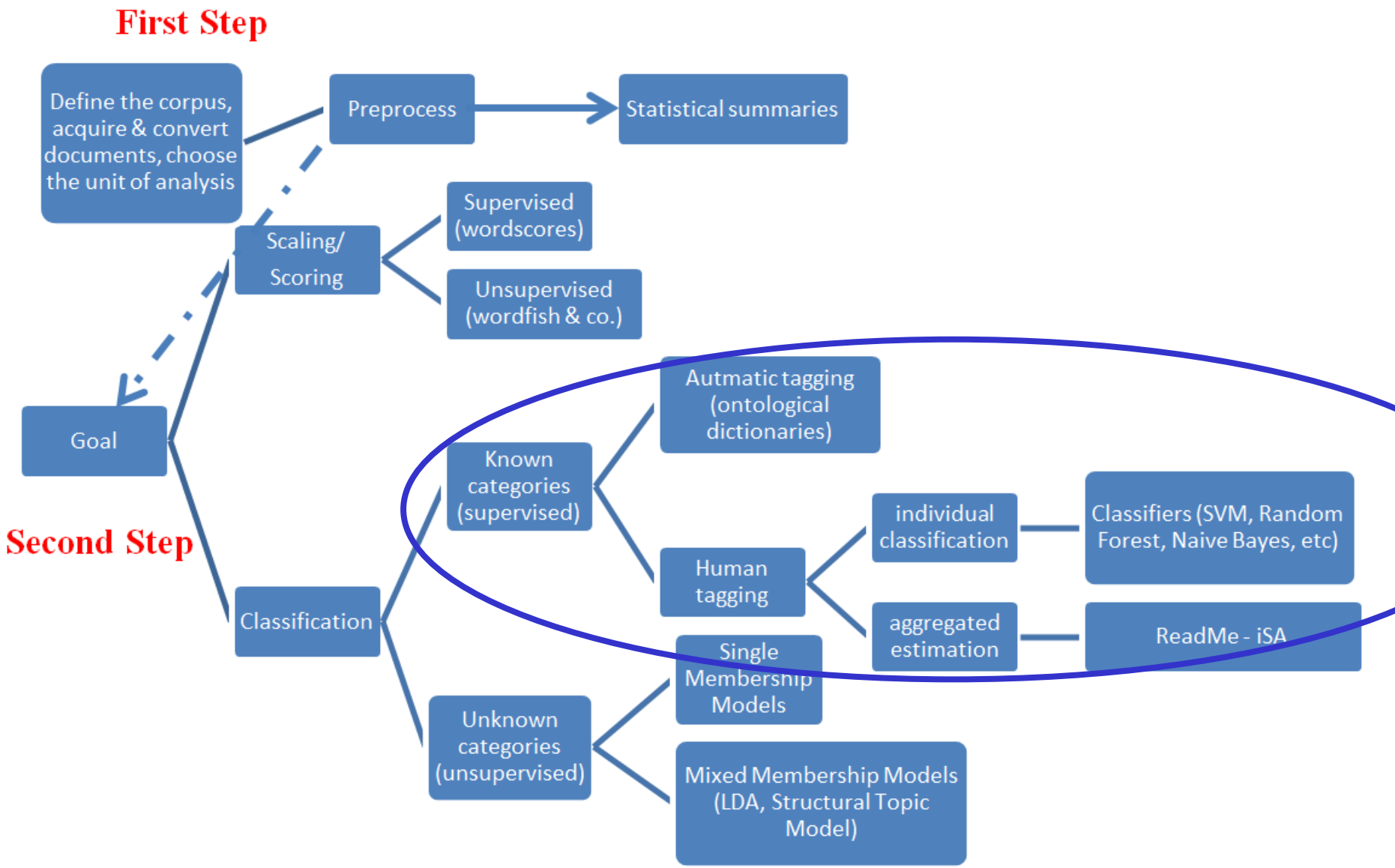
Applied Scaling & Classification Techniques in Political Science

Lecture 6

Dictionaries and an introdution to Supervised classification methods



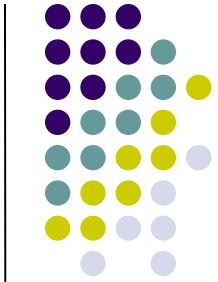
Our Course Map



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
- ✓ Olivella, Santiago, and Shoub Kelsey (2020). Machine Learning in Political Science: Supervised Learning Models. Luigi Curini and Robert Franzese (eds.), *SAGE Handbook of Research Methods is Political Science & International Relations*, London, Sage, chapter 56

Classification methods



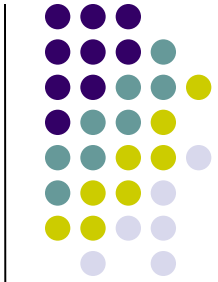
Classifying Documents into Known Categories

Assigning texts to **some known categories** is the most common use of content analysis methods in political science

For example, researchers may ask if local news coverage is positive or negative, if legislation is about the environment or some other issue area, if international statements are belligerent or peaceful, etc.

In each instance, the goal is to infer **either the category of each document**, the **overall distribution of documents across categories**, or **both**

Classification methods



Human-based methods for making these inferences are both time and resource intensive

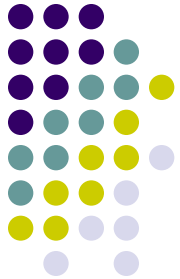
Automated methods can mitigate the cost of assigning documents to categories

There are **two broad groups of supervised classification methods** available according to the type of **tagging** (i.e., the assignation of a document to a given pre-defined category) **employed**:

We can have either

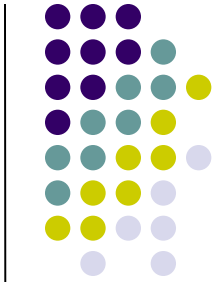
1) automatic tagging
2) human tagging

Automatic tagging



Dictionaries use the **relative rate** at which key words appear in a text to **classify documents into categories** or **to measure the extent to which** documents belong to particular categories

Human tagging



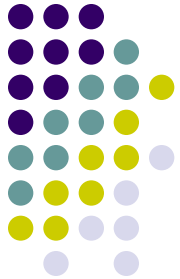
Supervised learning methods replicate the familiar manual coding task, but with a machine

First, human coders are used to classify a subset of documents into a predetermined categorization scheme

Then, this training set is used to train an automated method, which then classifies the remaining documents

Let’s start with automatic tagging…

Dictionary methods



Suppose the goal is to measure the **tone** (also called the "**sentiment"**) in newspaper articles: whether articles convey information positively or negatively about a given topic

A **dictionary to measure sentiment** is a list of words that are either dichotomously classified as positive («good», «fantastic», etc.) or negative («bad», «horrible», etc.) or contain more continuous measures of their content

You can then use that dictionary to identify **the tone of a document**: either positive or negative according to the relative number of words in that document identified by the dictionary as positive or negative ones

Dictionary methods

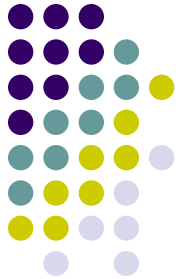
Formally, within a given dictionary Z each word *m* (m=1,….M) will have an associated score s_m

For the simplest measures, s_m =-1 if the word is associated with a negative sentiment and s_m =+1 if associated with a positive sentiment

If $N_i = \sum_{m=1}^{M} W_{im}$ words included in dictionary Z are used in document *i*, then dictionary methods can use such list of words to measure the sentiment for any document t_i as:

$$t_i = \sum_{m=1}^{M} \frac{s_m W_{im}}{N_i}$$

Dictionary methods



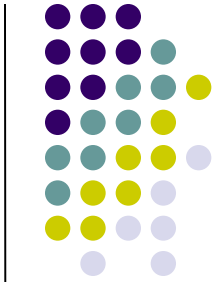
Scholars often use t_i as an approximately continuous measure of document sentiment, but it also can be used to classify documents into **sentiment categories** if a decision rule is assumed along with the dictionary method

Perhaps the simplest coding rule would assign all documents with $t_i > 0$ to a positive sentiment category and $t_i < 0$ to a negative sentiment

And if $t_i = 0$? Either neutral category or NC

Of course, all the words included in the texts you want to analyze **not also included** in the dictionary, will not provide any additional information for your classification aim

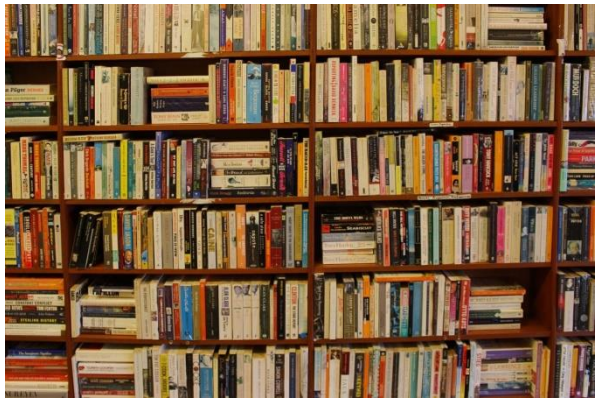
Dictionary methods

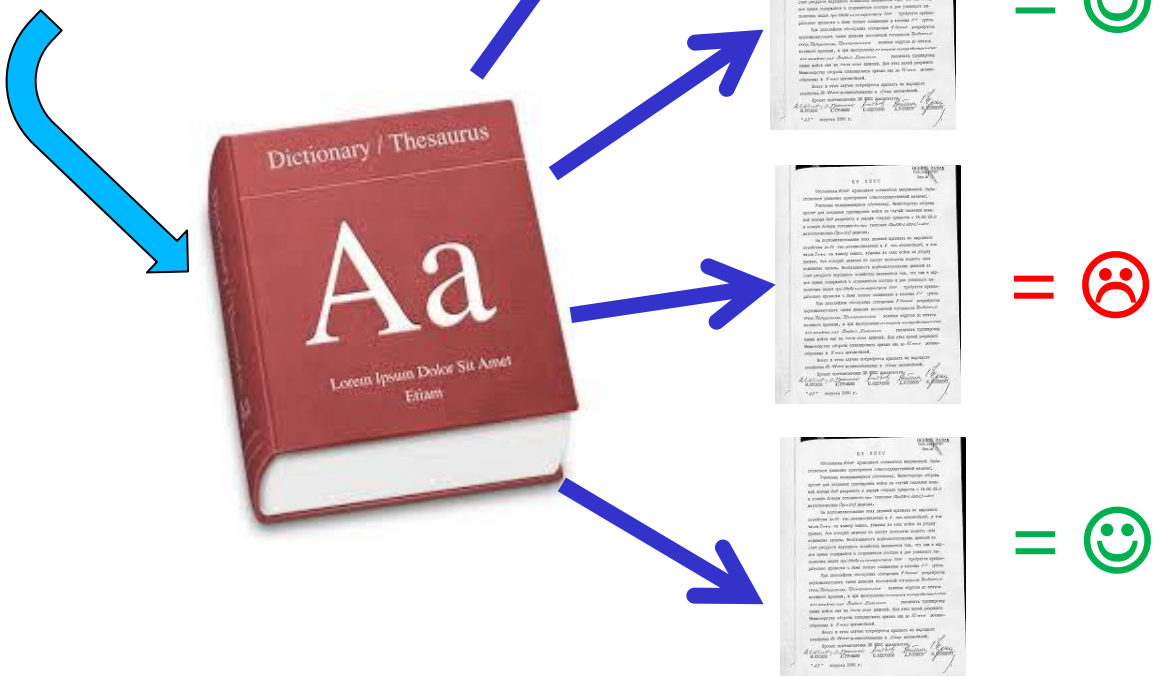


Sentiment analysis is just one type of analysis a dictionary method can perform

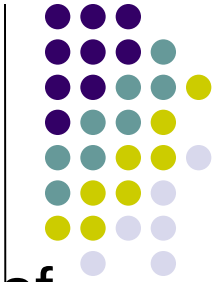
The general idea of dictionaries make them relatively easy and cheap to apply across a variety of problems: identify **words that separate categories** (for example *policy categories*) and measure **how often those words occur in texts**

Dictionary methods





Dictionary methods

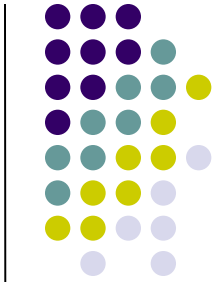


Using a dictionary can therefore **minimize** the amount of labor needed to classify documents (and this is attractive!) BUT…

…the difficulty lies in constructing a dictionary so that all relevant terms are included (**no false negatives**, i.e., terms we should have included in the dictionary cause they are relevant given our research topic, but failed to do so), but no irrelevant or wrong terms are (**no false positives**, i.e., terms we have included in the dictionary but should not have, being them irrelevant given our research topic)

This is not that easy!

Dictionary methods



…the challenges of using a dictionary:

For dictionary methods to work well, the scores attached to words must closely align with **how** the words are used in a particular context

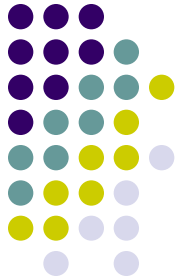
If a dictionary is developed for a **specific application**, then this assumption should be easy to justify

But when dictionaries are **created in one substantive area and then applied to another**, serious errors can occur

Language do **change across topics**!

For example, a word like `cancer` may have a positive connotation in a health-care company, but negative in many other contexts

Dictionary methods



Moreover, dictionary methods work pretty well when you study texts that use a **standardized language** (i.e., legal text!).

In other contexts, things become more complex…

Dictionary methods

…language in fact evolves continuously: one cannot code all possible semantic rules (double meaning sentences, specific jargons, neologisms, irony) unless reading the posts!!!

Dictionary methods

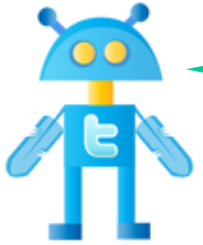
On the other side, counting the number of positive and negative terms in a sentence may lead to **paradoxical effects**

Dictionary methods

*"This movie has **good** premises. Looks like it has a **nice** plot, an **exceptional** cast, **first class** actors and Stallone gives his **best**. But it sucks"*

5 POSITIVE TERMS VS 1 NEGATIVE

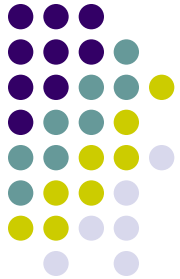
"What a nice rip-off"



50% *positive* & 50% *negative*
=
misclassification

100% *negative*
=
no misclassification

Dictionary methods



Dictionaries, therefore, should be used with **substantial caution**, or at least coupled with **explicit validation**

The problem is that quite often measures from dictionaries are **rarely validated**

Rather, standard practice in using dictionaries is to assume the measures created from a dictionary are correct and then apply them to the problem

The consequence of **domain specificity and lack of validation** is that most analyses based on dictionaries are built on shaky foundations

R pakcages to install

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

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

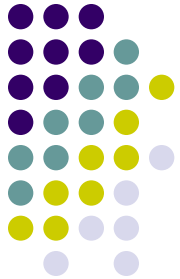
install.packages("plotly", repos='http://cran.us.r-project.org')

install.packages ("reshape2", ='http://cran.us.r-project.org')

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

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

Supervised Learning vs. Dictionary methods

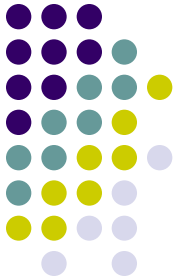


Dictionary methods:

- ✓ 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 **via human intervention**

Supervised Learning (classification) Methods

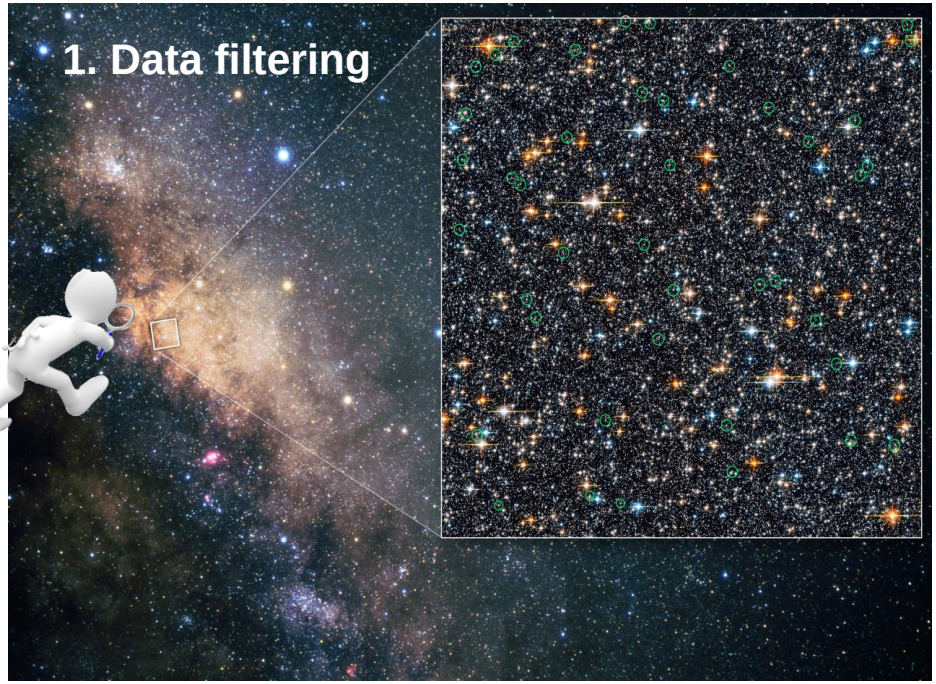


The idea of supervised learning is simple: human coders categorize a set of documents (the "**training-set" or "labelled-set"**) by hand

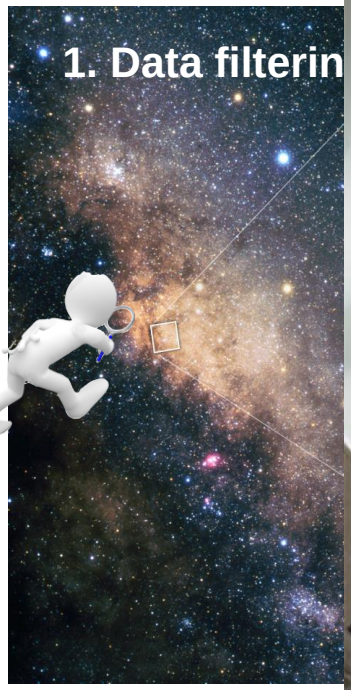
The algorithm "learns" how to sort the documents into categories using the **training set and words**

Then, it classifies the remaining set of document not classified by hand (the **"test-set"** or **"unlabelled set"**) using the characteristics of the documents to place them into the categories

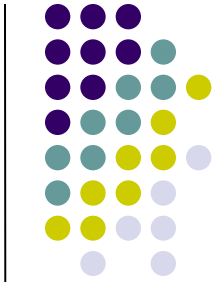
A three-steps procedure



A three-steps procedure



Supervised Learning Methods



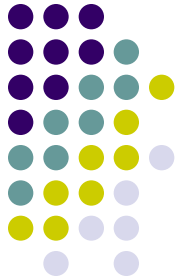
This approach to classification has **three major advantages** over dictionary methods:

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 Methods



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

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

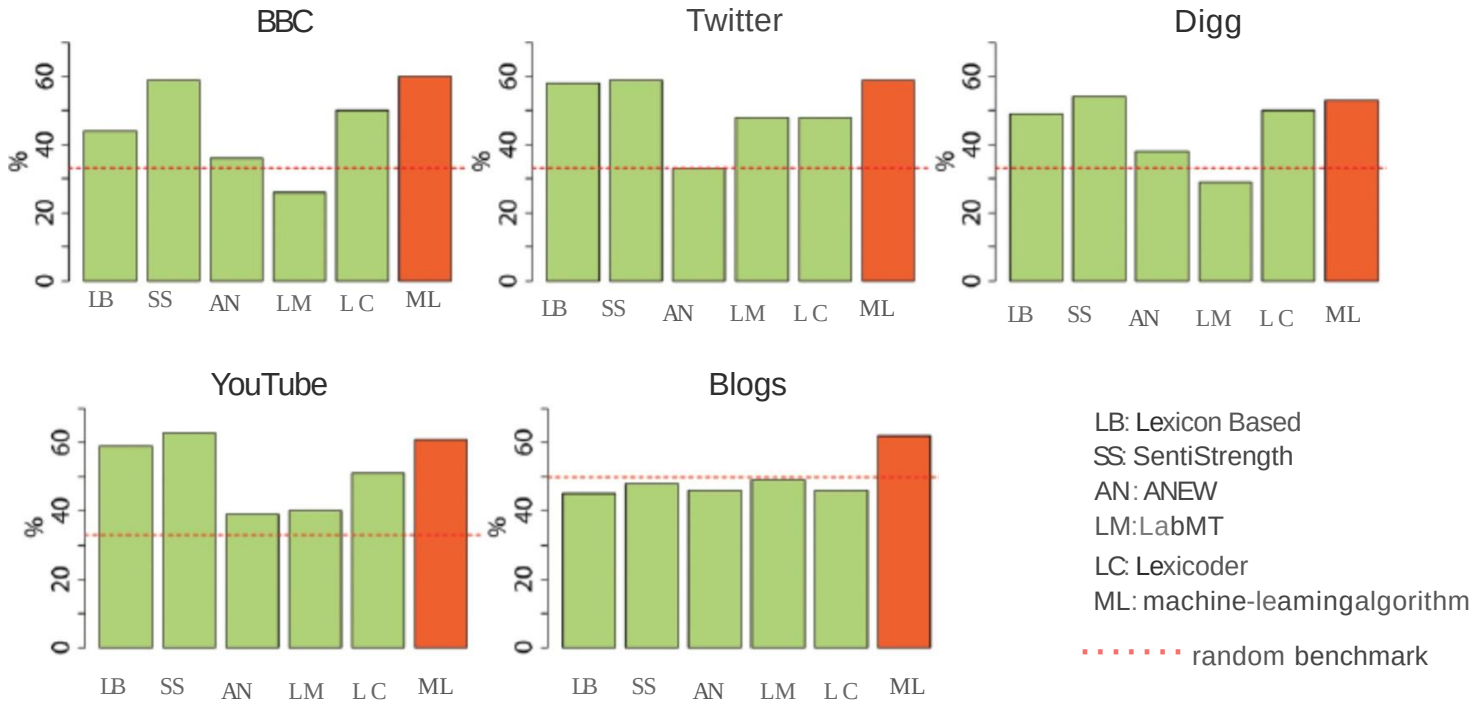
Third, supervised learning methods are much easier to validate, with clear statistics that summarize model performance (see below)

Supervised Learning vs. Dictionary methods

By construction, supervised learning methods will outperform dictionary methods in classification tasks, as long as training sample is large enough

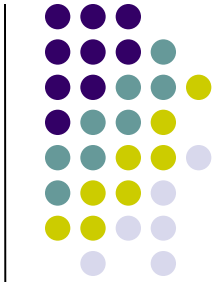
And indeed…

Supervised Learning vs. Dictionary methods



Source: Gonzalez-Bailn and Paltoglou (2015)

Constructing a training set



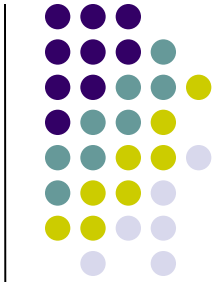
For supervised problems, the researcher is aiming to classify documents into a set of known or assumed categories based upon rules or information that can be learned from the training set

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



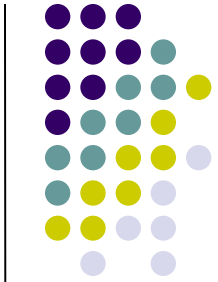
(1) creating and executing a coding scheme:

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

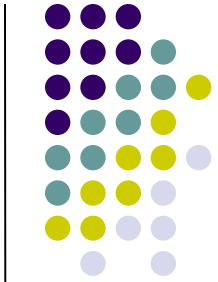


(1) creating and executing a coding scheme:

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:**

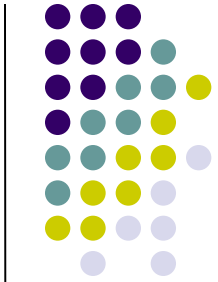
Almost all (but not all…) classification methods implicitly assume that the **training set is a random sample** from the population of documents to be coded

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

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

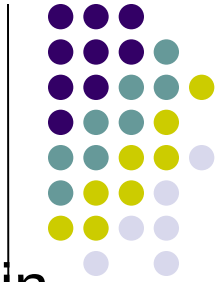
Constructing a training set



(2) **sampling documents:**

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

Classification methods



Supervised classification differs from supervised differs in that scaling aims to estimate a position on a latent dimension, while **classification aims to estimate a text's membership in a latent class**

The two tasks differ in how greedily they demand input data in the form of more features and additional documents

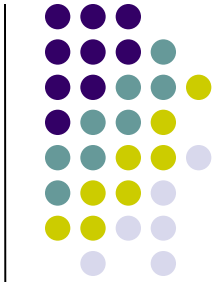
Typically, classification tasks can be improved by a**dding more training data**, while scaling tasks rest first and foremost on the quality (not the quantity) of the reference texts (as we have already discussed!!!)

Constructing a training set

(2) **sampling documents:**

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

Constructing a training set

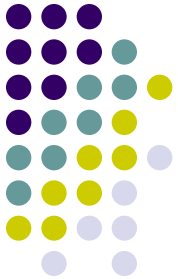


(2) **sampling documents:**

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



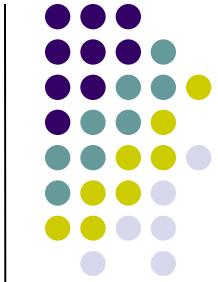
(2) **sampling documents:**

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

The feasibility of achieving an optimal training/test split for a given research question will in part depend on the size of the corpus….

That is, some research questions emphasizing particularly complex or difficult-to-detect classes, which in turn require larger training sets, may be best served by selecting a large corpus at the outset and allowing flexibility in achieving the appropriate training/test split

Constructing a training set



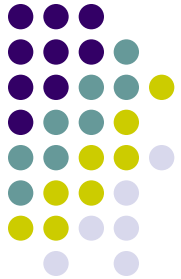
(3) **checking human-tagging reliability:**

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**!!!

Applying a supervised learning model

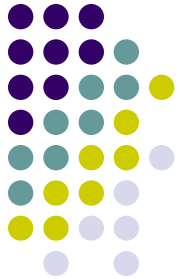


After hand classification is complete, the hand-labeled documents are used to train the supervised learning methods to learn about the test set – either:

a) classifying the **individual documents** into categories

b) measuring the **proportion of documents** in each category

Applying a supervised learning model

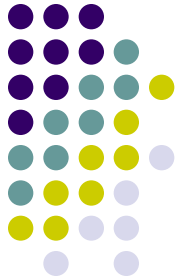


There is an important **theoretical (and statistical) difference** (and we will discuss about it – hopefully later!) between individual and proportional classification:

- ✓ for some social science application, only the proportion of documents in a category is needed, not the categories of each individual document
- ✓ The opposite happens in some other application

Once again, there is no “a best method” out there. It depends on your research topic (remember the 4 rules!)

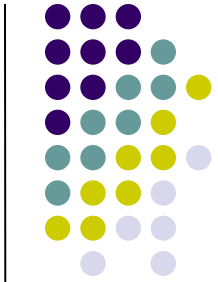
Applying a supervised learning model



Despite the fact that the methods to do supervised classification are diverse, they share a common structure that usefully unifies the methods

Let's first to discuss about **individual supervised classification methods** (i.e., **machine learning algorithms** applied to *text classification*)

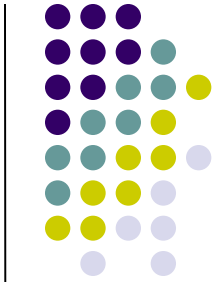
Machine learning



Machine learning is defined as the “field of study that gives computers the ability to learn without being explicitly programmed” (Samuel 1959)

In this context “learning” can be viewed as the use of statistical techniques to enable computer systems to progressively improve their performance on a specific task from data without being explicitly programmed (Goldberg and Holland 1988)

Machine learning



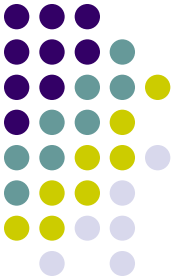
To be able to learn how to perform a task and become better at it, a machine should…

- ✓ …be provided with a set of example information (inputs) and the desired outputs. The goal is then to learn a general rule that can take us from the inputs to the outputs

In our case, our aim is to do *text classification*, i.e., to map a set of inputs (e.g., documents) to a predicted class as the output

Therefore, **machine learning algorithms** (when dealing with text classification methods) refer to those techniques that involve *individual* classification of the data in the *test set* given a pre-coded *training set*

Applying a supervised learning model

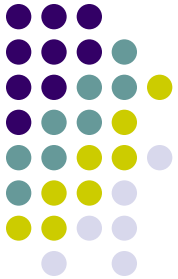


Suppose there are N_{train} documents (i=1,…, N_{train}) in our training set and that we have pre-defined K categories (k=1,…K) for our classification, such as positive, negative, neutral in the case of a sentiment analysis

Each document i's category is then represented by $Y_i \in (C_1, C_2, .., C_K)$ and the entire training-set is represented as $\mathbf{Y}_{train}=(Y_1,\ldots,Y_{Ntrain})$

$\mathbf{W}_{train}$ is the term-document matrix for N_{train}

Applying a supervised learning model



Each supervised learning algorithm assumes that there is some (unobserved) function that describes the (true) relationship between the words and the labels in the training-set:

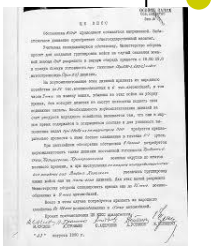
$$\boldsymbol{Y}_{train} = \mathrm{f}(\boldsymbol{W}_{train})$$

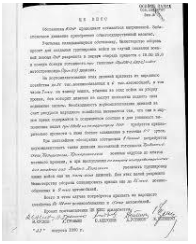
Each algorithm attempts to learn this relationship by estimating the "true" function *f* with $\hat{f}$ (the **classification function**)

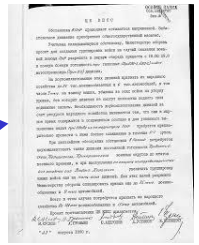
$\hat{f}$ is then used to infer properties of the test set (the unlabeled set), $\widehat{\boldsymbol{Y}_{test}}$ - **either** each document's category **or** the overall distribution of categories - using the test set's words $\boldsymbol{W}_{test}$:

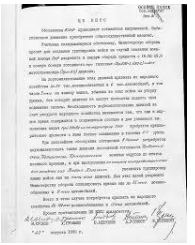
$$\widehat{\boldsymbol{Y}_{test}} = \hat{f}(\boldsymbol{W}_{test})$$

Supervised learning: individual classification

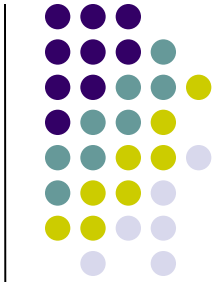








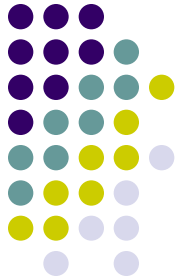
Beware of overfitting!



Summing up…

Supervised learning models **share the same goal**: learn the potentially complicated relationships that relate (combinations of) features x to the outcome of interest y in general, using information available in the set of observations for which the pair (x; y) is fully observed

Beware of overfitting!

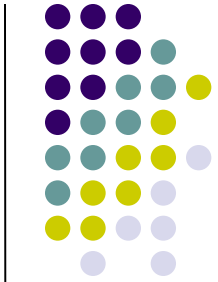


Still here it lies the riddle!

it is typically **easy** to learn even complicated relationships in-sample that is, relationships that are conditional on the training set

The goal, **however**, is to learn relationships for which the **expected generalization error** (i.e. the error that can be expected to ensue when learned relationships are evaluated out-of-sample, on a random test set of observations not involved in the learning process) is low

Beware of overfitting!

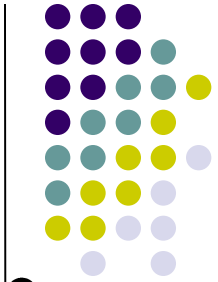


In fact, while it is always possible to arbitrarily reduce training error (i.e. error as computed using the training sample) by making models arbitrarily complex…

…such **complexity** typically results in high expected generalization error, as models start to overt their training data (i.e. they start to pick up on idiosyncratic relationships that conditional on the set of observations used to train the models)…

…that is, a machine learning algorithm begins to **overfit the data**!

Beware of overfitting!



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

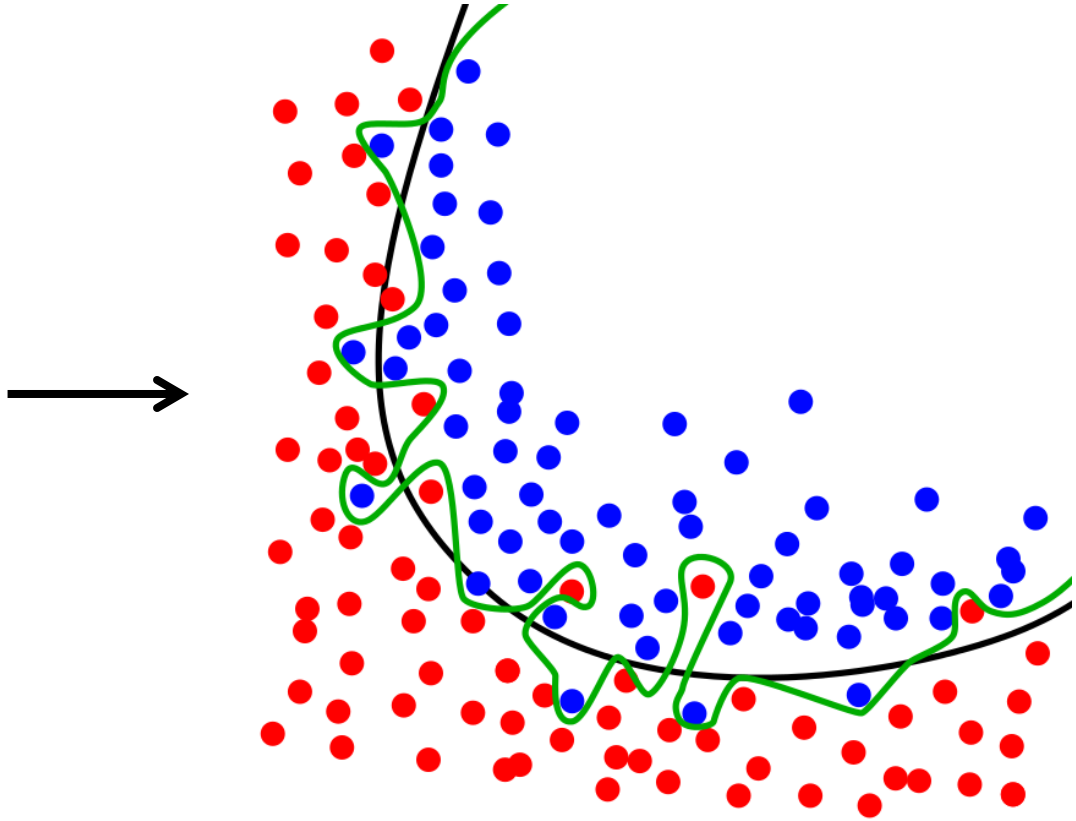
Overfitting is the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore **fail** to fit additional data or predict future observations reliably

An overfitted model is a statistical model that contains more parameters than can be justified by the data

Beware of overfitting!

Two examples of overfitting

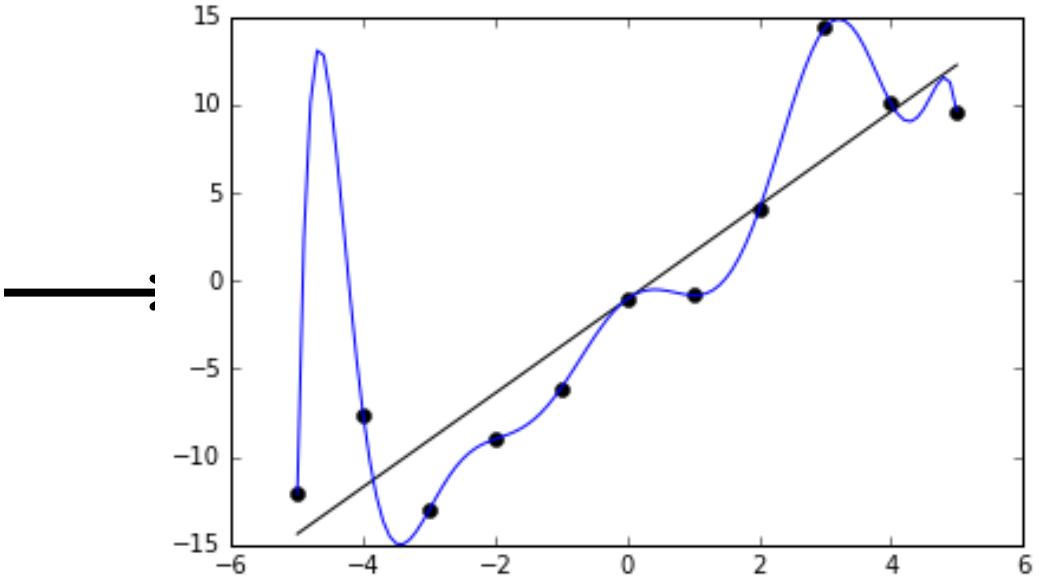
While the green line best follows the training data, it is too dependent on that data and it is likely to have a higher error rate on **new unseen data**, compared to the **black** line



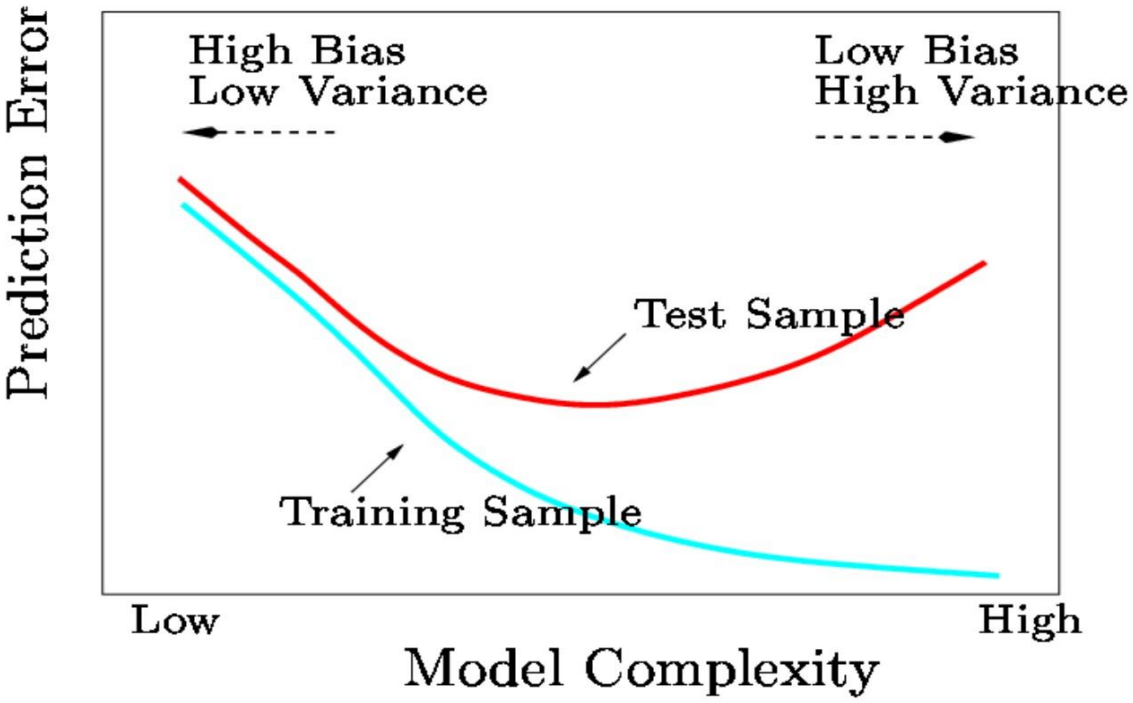
Beware of overfitting!

Two examples of overfitting

Although the polynomial function (the blue line) is a perfect fit, the linear function can be expected to generalize better beyond the fitted data!

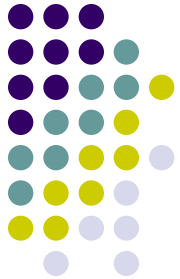


Beware of overfitting!



- ✓ 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
- ✓ In-sample performance better than out-of-sample performance

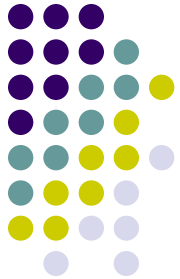
And so? Which solution?



Accordingly, and since the ultimate goal of supervised learning is to find generalizable patterns of association, models are typically subject to some from of regularization - typically in the form of a constraint that pushes the model toward parsimony - and are **selected based on their ability to generate good out-of-samples predictions**

Clearly, it is impossible to evaluate a model's performance on the universe of unsampled test instances, so an approximate measure of performance must be devised

And so? Which solution?



Although several approaches are viable, none is more commonly used than **cross-validation** (CV) - the exercise of further splitting the training data into a training set and a validation set (used to evaluate predictive accuracy, but omitted from the learning phase)

We will discuss a lot about this later on!