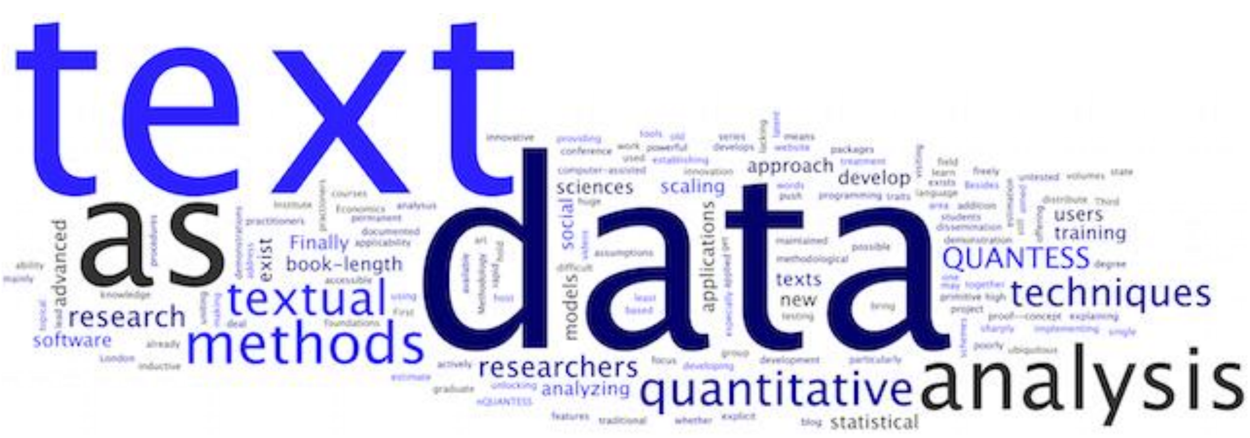
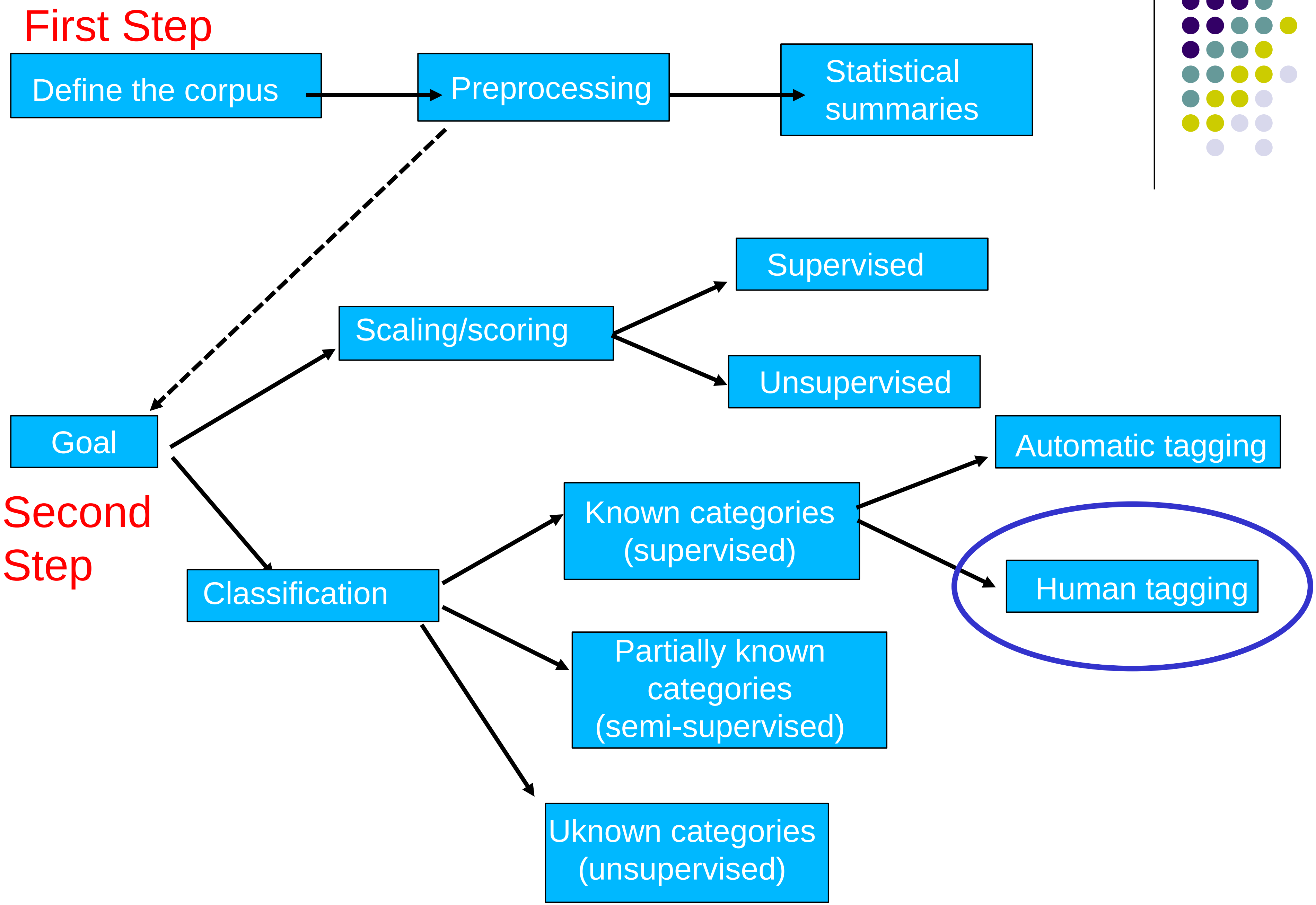
Applied Scaling & Classification Techniques in Political Science

Lecture 6 Part 2

Supervised classification methods with human tagging: an introduction




First Step
Define the corpus
Preprocessing
Statistical summaries
Supervised
Scaling/scoring
Unsupervised
Goal
Automatic tagging
Known categories (supervised)
Second Step
Human tagging
Classification
Partially known categories (semi-supervised)
Uknown categories (unsupervised)

References

- ✓ 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 (classification) Methods

The idea of supervised learning is simple: human coders categorize a set of documents (the "**training-set" or "labelled-set"**) by hand into a set of pre-defined categories (such as positive, negative, neutral for example)

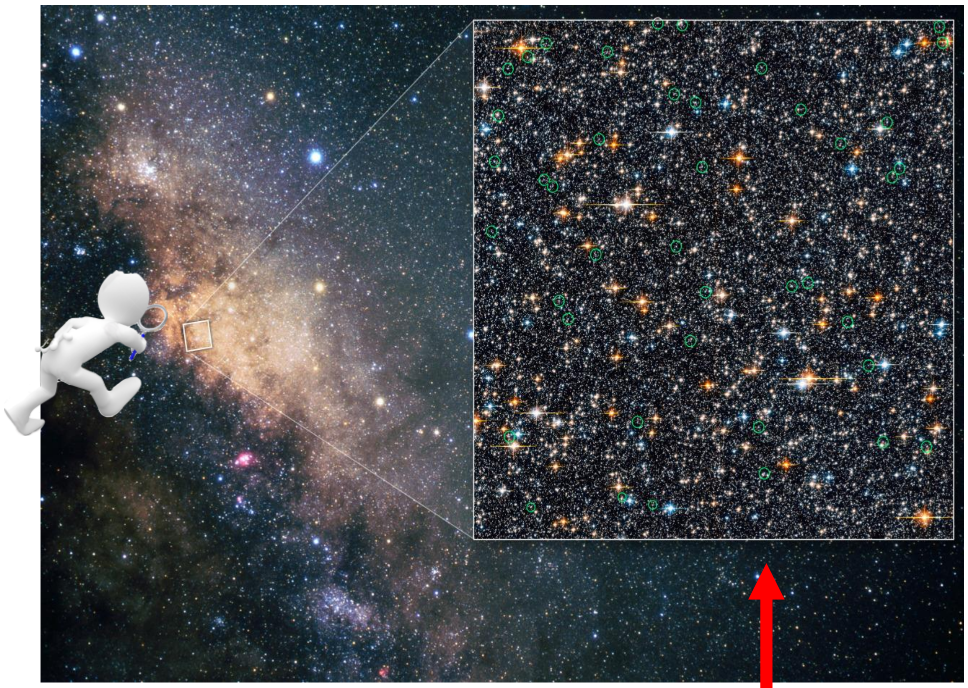
The algorithm "learns" how to sort the documents into categories using the **training set and words**

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

A four-step procedure

1. Data preparation: separating the training set from the test set in the corpus

2. Human classification
of the training set on a base of a list of pre-defined categories

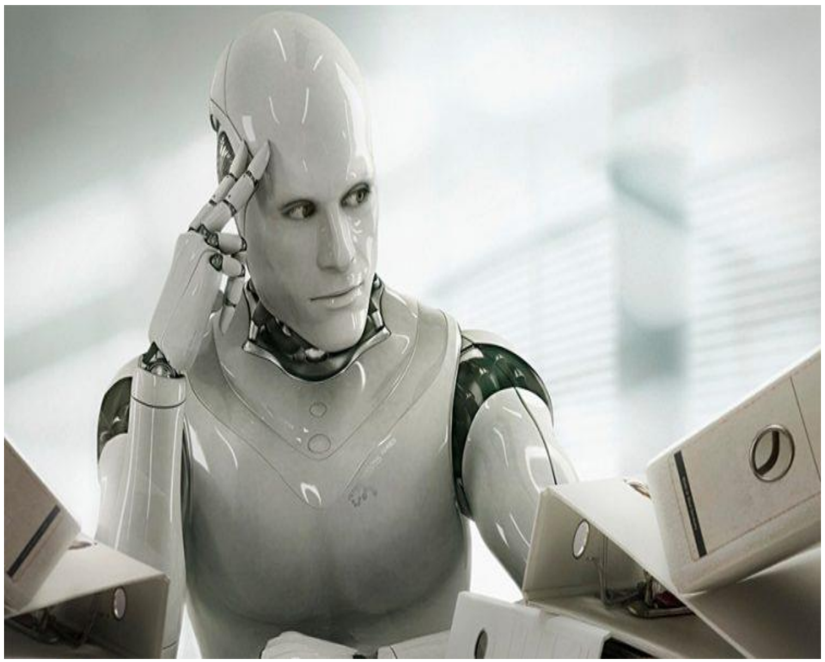


the training set

A four-steps procedure

3. Cogito ergo sum! The algorithm learns from the human classification done in the training set

4. Let's classify! The well-educated algorithm is now ready to classify all the texts in the test-set



The algorithms that we will employ belong to the Machine Learning class

Machine learning

Machine learning is defined as the "field of study that gives computers the ability to learn [NOT TO THINK!] 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

Machine learning

In our case, our aim is to do *text classification*

Therefore, **machine learning algorithms** (when dealing with text classification methods) refer to those techniques that learn how to map a set of inputs (e.g., features within documents) to a predicted class as the output in a pre-coded *training set (via human intervention)* before classifying the data in the *test set*

Supervised Learning (classification) Methods

- Despite the fact that the methods to do supervised classification are diverse, they share a **common structure** that usefully unifies the methods
- The `dfm` *S* representing our corpus has *N* rows and *L* columns, with each document *i* being represented by a vector S_i of length *L*
- The value of each element of the vector S_i as we well know may either be the frequency with which that feature appeared in document *i*, or a binary value – 1 if the feature appeared at all, 0 if it did not
- Of course, more than one text in the corpus can be represented by the same vector S_i

Supervised Learning (classification) Methods

This `dfm` is then divided into two subsets. One of length *n*, which is called the *training set*, and the remaining one of size *N-n*, the *test set*

We denote by (D_j,S_i) the couple that contains the coded value (label) D_j for text S_i

Clearly, D_j is a value in *D* for the texts in the training set (such as positive, negative, neutral, if you think about a Sentiment classification) and "NA" for the uncoded texts in the test set

The texts in the training set are assumed to be classified by humans without error (we will return to this point in the next weeks…)

Supervised Learning (classification) Methods

Any machine learning algorithm will then try to predict the category D_j to which a given document *i* belongs given that its features are represented by the vector S_i

This model can be represented as $P(D|S)$ – for a given document *i* and set of categories (labels) *D*, the algorithm will find the specific value of D_j which maximises the conditional probability $P(D_{j=0,1,\ldots,D}|S_i)$

Supervised Learning (classification) Methods

In a (very) naive model, the elements of the matrix P(D|S) can be estimated by taking the proportion of all texts in the training set that are hand-coded as $D = D_j$ which have $S = S_i$ as feature vector

A text will then be assigned to the label j with the highest proportion

For example if $S = S_i$ appears in 5 texts in the training-set, once with D_j=*positive* and four-times with D_j=*negative*, then every text in the test-set $S = S_i$ will be assigned to D_j=*negative*

Any machine learning model does essentially the same thing in more sophisticated ways (as we will learn)

Supervised Learning (classification) Methods

Summing up…

All 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 (i.e., in the training-set)

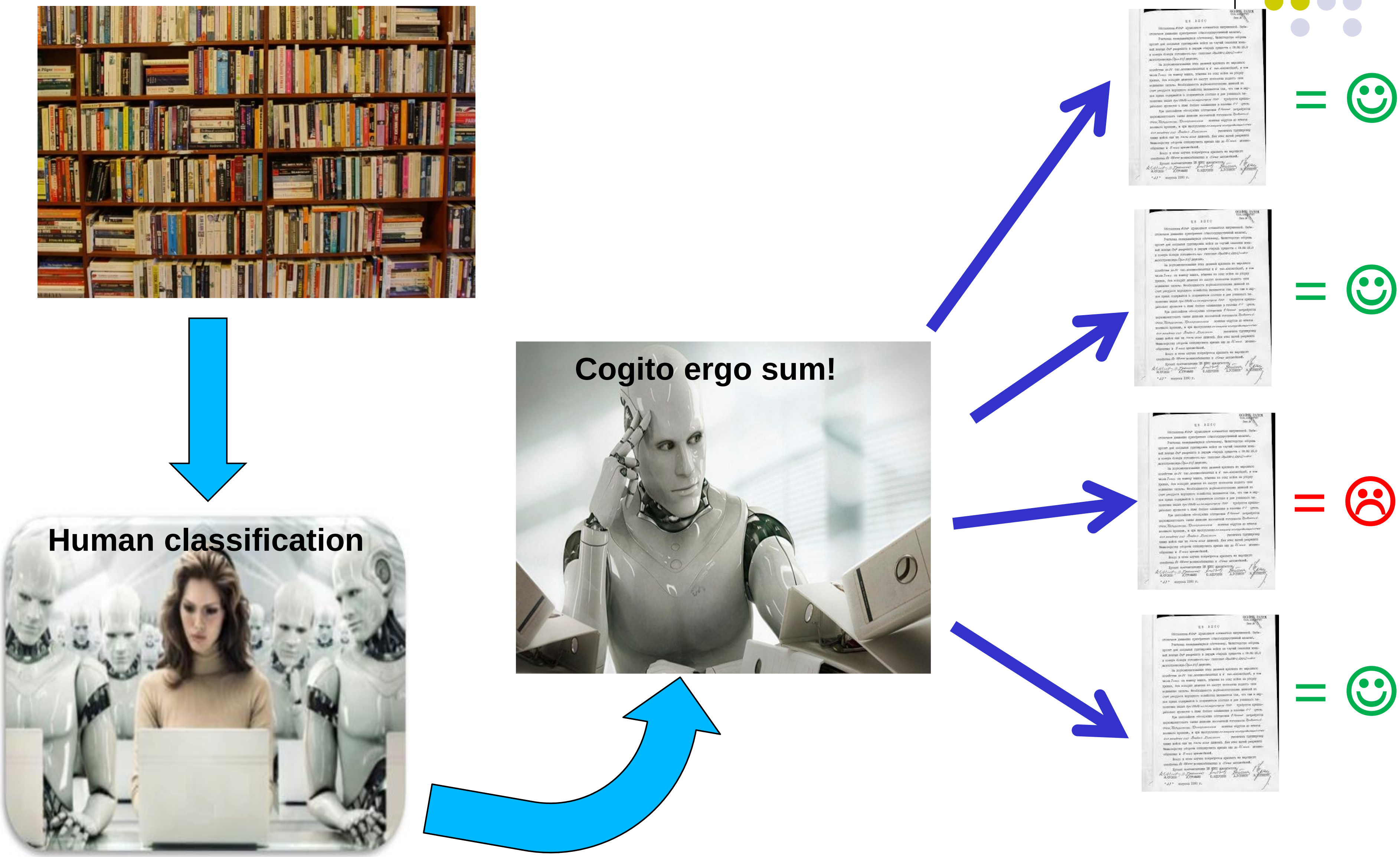
Supervised Learning (classification) Methods

More in details: each supervised learning algorithm assumes that there is some (unobserved) function that describes the (**true**) relationship between the words *S* and the labels *D* in the training-set

It then attempts to learn this relationship by approaching the "true" function with a **classification function** $\hat{P}(D|S)$ extracted from the subset of observations in the training set

$\hat{P}(D|S)$ is then used to infer properties (labels) of the test set, using the actual S_i (i.e., the features) in the test set for each document in the test set

Supervised learning



Supervised Learning (classification) Methods

From this point of view, the *Wordscores* approach to supervised scaling can be compared also to supervised ML

You have a training-set (reference texts) that have been labelled (on a continuous scale)

You have an algorithm that learns from such training-set the relationship between features and the reference scores

You have then a test-set (virgin texts) whose scores will be predicted by the now “trained” algorithm

Supervised Learning vs. Dictionary methods

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**

The feature space is thus likely to be both larger and more comprehensive than that used in a dictionary

Moreover, the weight of each single feature is not defined ex-ante (as when applying a dictionary), but it is discovered ex-post according to the corpus

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

Supervised Learning vs. Dictionary methods

Moreover, compared to dictionary methods:

Supervised learning is **necessarily domain specific** and therefore avoids the problems of applying dictionaries outside of their intended area of use

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

Finally, supervised learning methods are much easier to validate, with clear statistics that summarize model performance (as we will discuss)

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, that is, 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:

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

However remember: while a dictionary cannot compete with a classifier trained on a **representative and large enough training dataset**, in any given task (good) dictionaries may outperform a supervised learning model **if these conditions are not met**

Beware of overfitting!

We have just discussed how **machine learning algorithms** (when dealing with text classification methods) refer to those techniques that learn how to map a set of inputs (e.g., features within documents) to a predicted class as the output in a pre-coded *training set* before classifying the data in the *test set*

Beware of overfitting!

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

But our goal is a different one!

We want 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 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 of the training-data, rather than capturing the “true” property of the distribution from which such data arose)…

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

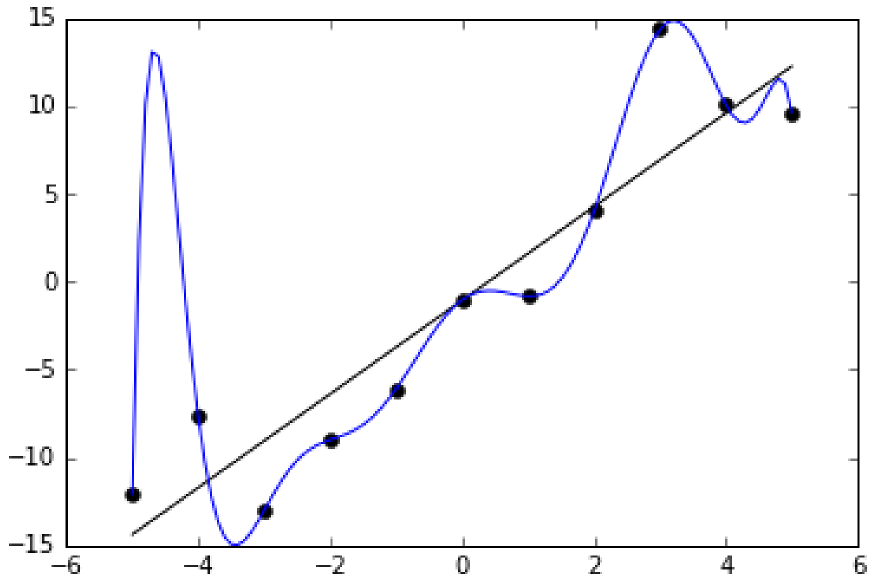
Beware of overfitting!

Overfitting is the production of an analysis that corresponds too closely to a particular set of (training) data, and may therefore **fails** to fit additional data or predict future observations (i.e., the test set) reliably

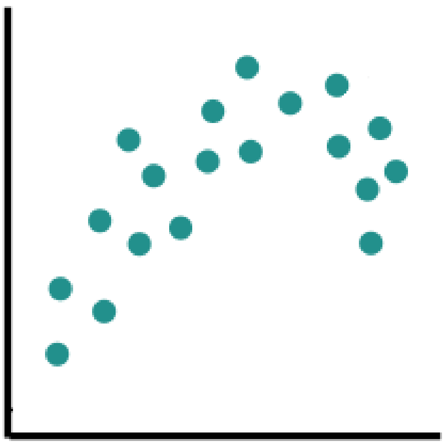
Beware of overfitting!

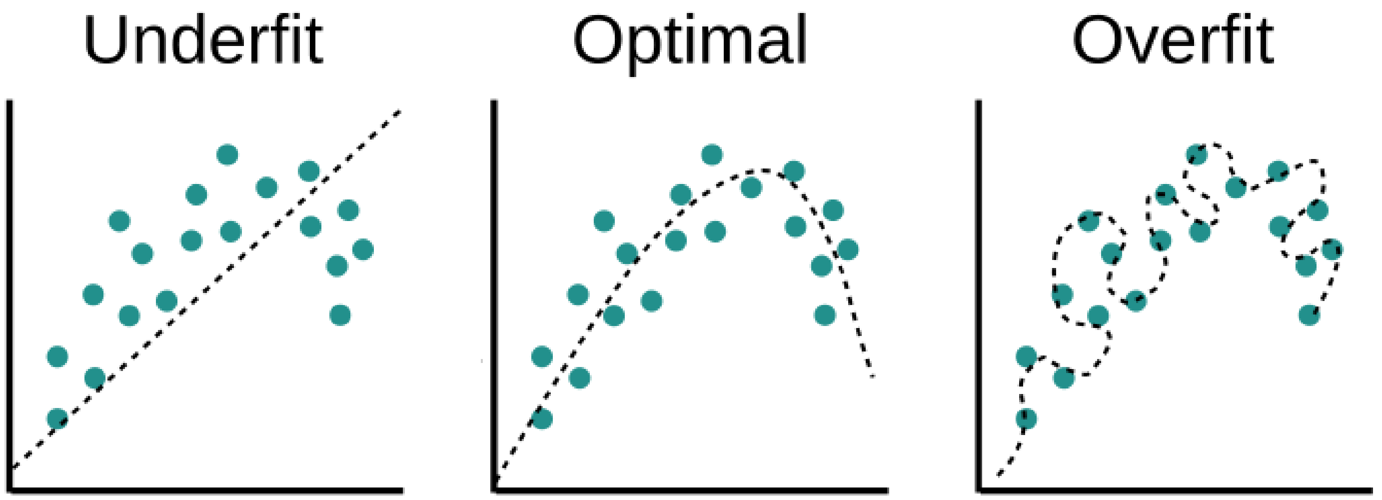
Although the polynomial function (the blue line) is a perfect fit, the linear function can be expected to generalize better beyond the fitted data!

→

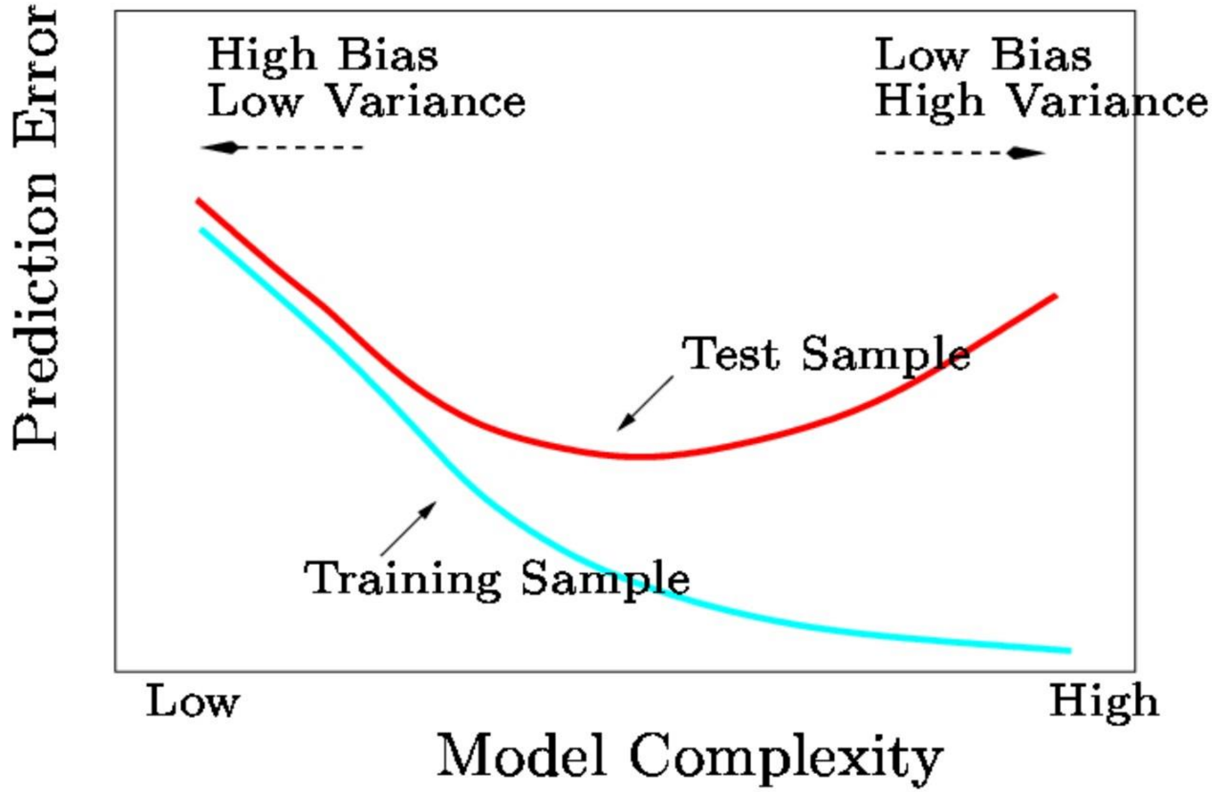


Beware of overfitting!





The Bias-Variance trade-off



- ✓ If a model is too complex, it describes noise rather than signal
- ✓ It will focus on features that perform well in training-set data but may not generalize, that is…
- ✓ In-sample performance will be (much) better than out-of-sample performance