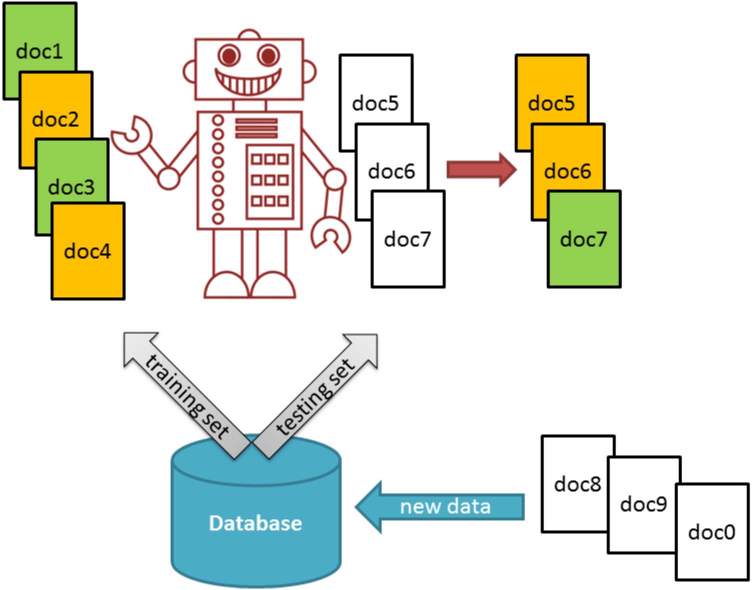
BINOMIAL CLASSIFICATION

Text Analytics - a Shortcut to Linguistic Evidence
@Center for Language Technology|University of Copenhagen

Kristoffer L Nielbo
knielbo@sdu.dk
knielbo.github.io

Department of History|University of Southern Denmark


doc1
doc2
doc3
doc4
doc5
doc6
doc7
doc5
doc6
doc7
training set
testing set
Database
new data
doc8
doc9
doc0

Classification

- Given labeled data (supervised learning), a classification algorithm will output a solution that categorizes new examples $\rightarrow$ associate labels with subsets of the data

- while clustering (unsupervised learning) searches for groups within the corpus, classification learns to map a collection of documents onto a categorical class values or labels $\rightarrow$ find mapping function

- iow. classification is *function approximation*

- data (features) with class values ($\sim$ labeled data), excellent opportunity to make use of metadata

- vast majority of models are *black box models*

Workflow preprocessing/selection $\rightarrow$ split $\rightarrow$ train $\rightarrow$ test $\rightarrow$ apply

Data Preprocessing normalizes linguistic data and removes highly frequent/infrequent terms in order to increase discriminatory power and reduce computational costs

– case-folding, removal of punctuation, and tokenization

– pruning and stopword filtering

– POS-tagging, stemming and lemmatization

```
"Dante passes through the gate of Hell, which bears an inscription ending with the famous phrase
'Abandon all hope, ye who enter here."

['Dante', 'passes', 'through', 'the', 'gate', 'of', 'Hell', 'which', 'bears', 'an', 'inscription',
 'ending', 'with', 'the', 'famous', 'phrase', 'Abandon', 'all', 'hope', 'ye', 'who', 'enter', 'here']

['Dante', 'passes', 'gate', 'Hell', 'bears', 'inscription', 'ending', 'famous', 'phrase', 'Abandon',
 'hope', 'ye', 'enter']

['dante', u'pass', 'gate', 'hell', u'bear', 'inscription', u'end', 'famous', 'phrase', 'abandon',
 'hope', 'ye', 'enter']
```

Feature Selection can be using a univariate statistical test (e.g., χ^2) and then selecting the k highest scores or we can estimate the mutual information between discrete variables (t and c) and "select k best":

$$IG(t,c) = \sum_{c' \in (c,\bar{c})} \sum_{t' \in (t,\bar{t})} P(t',c') \log \frac{P(t',c')}{P(t')P(c')} \tag{1}$$

For training a model, we need **labeled data** for establishing a ground truth to train up against

- metadata is readily available, e.g. author, genre, year of publication
- labels from an external source/databases, e.g. reviews, ratings, reads
- annotated/rated data (expert or raters)

Examples of classifiers (ML algorithms) for text classification (binary and multiclass problems):

Naive Bayes, probabilistic learning algorithm based on Bayesian decision theory

LogitBoost, boosting algorithm that implements forward stagewise modeling to form additive *logistic regression*

Support Vector Machines, popular algorithm that use linear models to seperate a nonlinear space

Naive Bayes

A simple and very popular probability learning model that can be implemented very efficiently

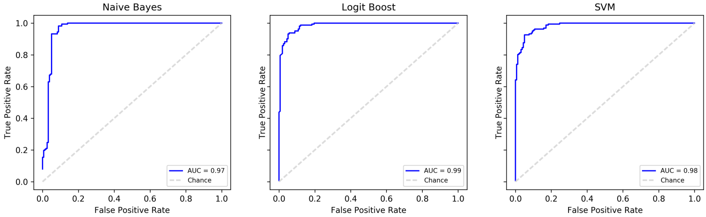
The probability of a document d being in class c, $P(c \mid d)$ is computed as:

$$P(c \mid d) \propto P(c) \prod_{i=1}^{m} P(t_i \mid c) \tag{2}$$

and the class of a document d is then computed as:

$$c_{MAP} = arg\ max_{c \in \{c_1, c_2\}} P(c \mid d) \tag{3}$$

Naive assumption that the presence/absence of a feature is completely independent of other features.



Construct a pipeline that trains multiple models in order to identify the optimal classifier given the task

If the model gets enough data, it can basically memorize the data set (overfitting) $\rightarrow$ need to test the model on held-out data

Validation when building a predictive model, we need a way to evaluate the capability of the model on unseen data

- conventional validation (data split)
- cross validation
- bootstrap

Evaluation of performance is computed by comparing the classifier's predictions to ground truth

Performance metrics summerize classifier performance and are used to select between a set of classifiers
– most metrics are developed for binary classification problems

a Confusion matrix , C, is a contingency table that describes performance on training and/or testing data
– C is such that $C_{i,j}$ is equal to the number of observations known to be in group i but predicted to be in group j.

		PREDICTED	
		positive	negative
TRUE	positive	$c_{1,1}$	$c_{1,2}$
	negative	$c_{2,1}$	$c_{2,2}$

SDU

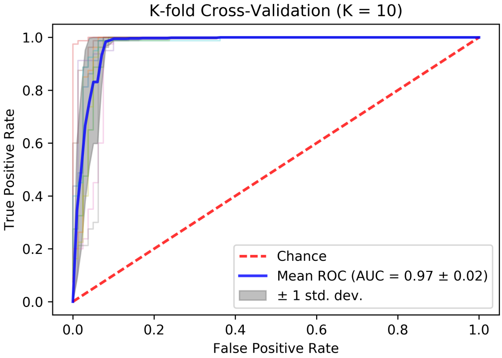
		PREDICTED	
		positive	negative
TRUE	positive	TP	FN
	negative	FP	TN

- **TP** Correctly assigns positive class membership
- **TN** Correctly rejects class membership
- **FP** Fail to rejects class membership (Type I error)
- **FN** Rejects class membership incorrectly (Type II error)

True Positive Rate (TPR, *sensitivity*, *recall*): $\frac{TP}{TP+FN}$

False Positive Rate (FPR, false alarm rate): $\frac{FP}{FP+TN}$

Construct a Receiver Operating Characteristics (ROC) graph from TPR and FPR at different thresholds for assigning an object to a class (positive)



We can then compute the Area Under the Curve of the ROC graph as a measure of accurracy

– the probability that our classifier will rank a randomly chosen positive instance higher that a randomly chosen negative instance

– no realistic classifier should have an AUC < 0.5

we sample 30 passages of the KJV Bible labeled with collection data (NT: New Testament OT: Old Testament) with an equal distribution

confusion matrix for binary classification problem:

	NT	OT
NT	10	5
OT	7	8

$\sim$ two raters/annotators (rows: ground truth; columns: classifier), then we are measuring their inter-rater reliability

15 are NT and 15 are OT, but model classified 17 as NT ad 13 as OT

$$Observed\ Accuracy : \frac{10+8}{30} = 0.6$$

$$Expected\ Accuracy : \frac{\frac{(10+5)\times(10+7)}{30} + \frac{(7+8)\times(5+8)}{30}}{30} = \frac{8.5+6.5}{30} = 0.5$$

50% will always be the random baseline in binary classification, when either rater/annotater classifies each class with the same frequency

Accuracy (*observed*) measures in how many cases the predicted class conformed with the correct class:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

	NT	OT
NT	10	5
OT	7	8

$$Accuracy = \frac{(10 + 8)}{10 + 5 + 7 + 8} = 0.6\ (60\%)$$

Cohen's κ compares the observed (p_o) to the expected chance-level (p_e) agreement on a classification problem:

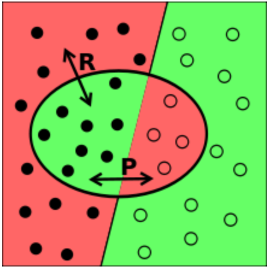
$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{5}$$

"how closely a classifier matches ground truth, controlling for the accuracy of a random classifier"

$$\kappa = \frac{0.6 - 0.5}{1 - 0.5} = 0.2$$

Interpretation: 0-0.2 slight, 0.21-0.4 fair, 0.41-0.6 moderate, 0.61-0.8 substantial, 0.81-1 perfect.

better than *Accuracy*, because random chance in included



← relevant objects (e.g., ham)

→ irrelevant objects (e.g., spam)

⬭ objects classified with relevant class label

ERROR

CORRECT

Precision: fraction of retrieved instances that are relevant

$$P = \frac{TP}{TP + FP} \tag{6}$$

Recall: fraction of relevant instances that are retrieved

$$R = \frac{TP}{TP + FN} \tag{7}$$

P and *R* are inversely related. Identify balance through a Precision-Recall curve.

ex. Precision measures the number of selected passages that are relevant, i.e., how certain are we that a classified passage is correctly classified ($\sim$ how many time did the model positively predict a class):

	NT	OT
NT	10	5
OT	7	8

$$\frac{TP}{TP + FP} = \frac{10}{10 + 7} = 0.59$$

For each class, how many of the passages that got the NT label should have gotten it?

ex. Recall measures the number of relevant passages that are selected, i.e., how good is the classifier at detecting verses within a given class:

	NT	OT
NT	10	5
OT	7	8

$$\frac{TP}{TP + FN} = \frac{(10)}{10 + 5} = 0.67$$

For each class, how many passages that should have gotten the NT label actually got it - how many were missed?

The F_1-score is a composite measure of a classifier's accuracy

$$F_1 = 2 \times \frac{\textit{percision} \times \textit{recall}}{\textit{precision} + \textit{recall}}$$

$$\text{biblical } F_1 : 2 \times \frac{0.59 \times 0.67}{0.59 + 0.67} = 0.63$$

F_1 is the harmonic mean of precision and recall.