Text Classification

[DAT640] Information Retrieval and Text Mining

Krisztian Balog
University of Stavanger

2024

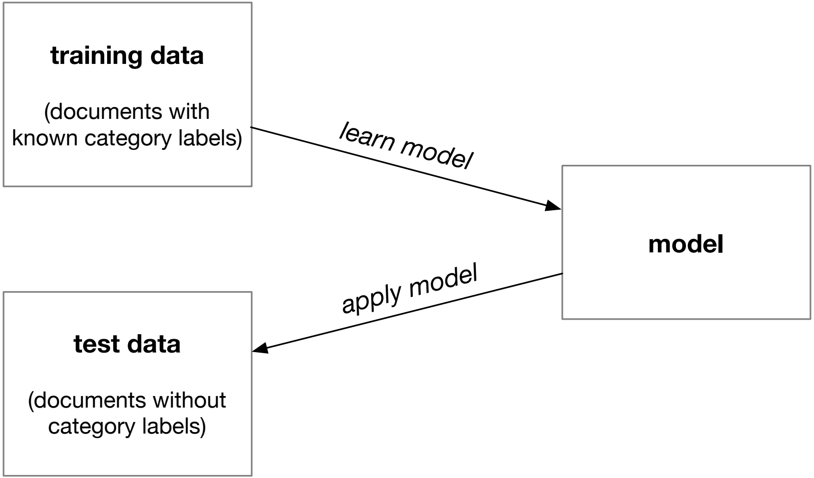
Table of contents

Text classification

Text classification

- **Classification** is the problem of assigning objects to one of several predefined categories
  - One of the fundamental problems in machine learning, where it is performed the basis of a training dataset (instances whose category membership is known)
- In **text classification** (or **text categorization**) the objects are text documents
- Binary classification (two classes, $0/1$ or -/$+$)
  - E.g., deciding whether an email is spam or not
- Multiclass classification (n classes)
  - E.g., Categorizing news stories into topics (finance, weather, politics, sports, etc.)

General approach



Formally

- Given a training sample $(\boldsymbol{X}, y)$, where $\boldsymbol{X}$ is a set of documents with corresponding labels y, from a set $\boldsymbol{Y}$ of possible labels, the task is to learn a function $f(\cdot)$ that can predict the class $y' = f(x)$ for an unseen document x.

Families of approaches

- **Feature-based approaches** (“traditional” machine learning)
- Neural approaches (“deep learning”)

Features for text classification

- Use words as features (**bag-of-words**)
  - Words will be referred to as **terms**
- Values can be, e.g., binary (term presence/absence), integers (term counts), or reals (weighted term importance)
- Documents are represented by their **term vector**
- **Document-term matrix** is huge, but most of the values are zeros; stored as a sparse matrix

	t_1	t_2	t_3	$\dots$	t_m
d_1	1	0	2		0
d_2	0	1	0		2
d_3	0	0	1		0
$\dots$					
d_n	0	1	0		0

Document-term matrix

Additional features for text classification

- Descriptive statistics (avg. sentence length, length of various document fields, like title, abstract, body,...)
- Document source
- Document quality indicators (e.g., readability level)
- Presence of images/attachments/JavaScript/...
- Publication date
- Language
- ...

Text classification evaluation

Evaluation

- Measuring the performance of a classifier
    - Comparing the predicted label y' against the true label y for each document in some set dataset
- Based on the number of records (documents) correctly and incorrectly predicted by the model
- Counts are tabulated in a table called the **confusion matrix**
- Compute various **performance measures** based on this matrix

Text classification evaluation

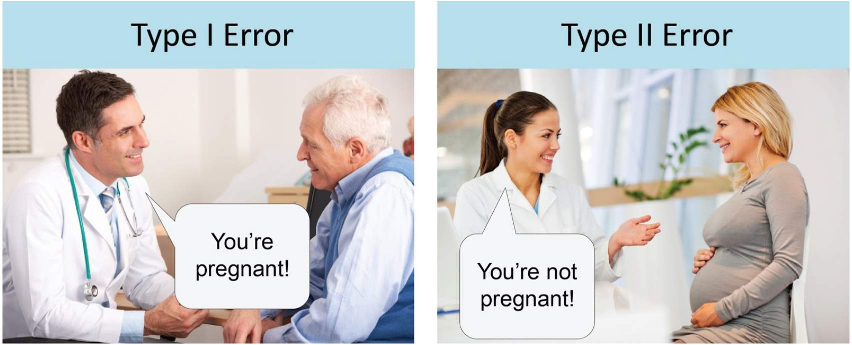
- Evaluating binary classification
- Evaluating multiclass classification
- Model development

Confusion matrix

		Predicted class	
		negative	positive
Actual class	negative	true negatives (TN)	false positives (FP)
	positive	false negatives (FN)	true positives (TP)

- False positives = Type I error ("raising a false alarm")
- False negatives = Type II error ("failing to raise an alarm")

Type I vs. Type II errors[1]



[1]Source: `https://www.analyticsindiamag.com/understanding-type-i-and-type-ii-errors/`

Example

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

		predicted	
		-	+
actual	-		
	+		

Example

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

		predicted	
		-	+
actual	-	**2**	
	+		

Example

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

		predicted	
		-	+
actual	-	2	**1**
	+		

Example

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

		predicted	
		-	+
actual	-	2	1
	+	4	3

Evaluation measures

- Summarizing performance in a single number
- **Accuracy**
  Fraction of correctly classified items out of all items

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Error rate**
  Fraction of incorrectly classified items out of all items

$$ERR = \frac{FP + FN}{FP + FN + TP + TN}$$

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Evaluation measures (2)

- **Precision**
  Fraction of items correctly identified as positive out of the total items identified as positive

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

- **Recall** (also called Sensitivity or True Positive Rate)
  Fraction of items correctly identified as positive out of the total actual positives

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

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Evaluation measures (3)

- **F1-score**
  The harmonic mean of precision and recall

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

		predicted	
		-	$+$
actual	-	TN	FP
	$+$	FN	TP

Evaluation measures (4)

- **False Positive Rate (Type I Error)**
  Fraction of items wrongly identified as positive out of the total actual negatives

$$FPR = \frac{FP}{FP + TN}$$

- **False Negative Rate (Type II Error)**
  Fraction of items wrongly identified as negative out of the total actual positives

$$FNR = \frac{FN}{FN + TP}$$

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Example

		predicted	
		-	+
actual	-	TN=2	FP=1
	+	FN=4	TP=3

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5}{10} = 0.5$$

$$P = \frac{TP}{TP + FP} = \frac{3}{4} = 0.75$$

$$R = \frac{TP}{TP + FN} = \frac{3}{7} = 0.429$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} = \frac{2 \cdot {}^3\!/_4 \cdot {}^3\!/_7}{{}^3\!/_4 + {}^3\!/_7} = 0.545$$

Text classification evaluation

- Evaluating binary classification
- Evaluating multiclass classification
- Model development

Multiclass classification

- Imagine that you need to automatically sort news stories according to their topical categories

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Table: Categories in the 20-Newsgroups dataset

Multiclass classification

- Many classification algorithms are originally designed for binary classification
- Two main strategies for applying binary classification approaches to the multiclass case
  - One-against-rest
  - One-against-one
- Both apply a voting scheme to combine predictions
  - A tie-breaking procedure is needed (not detailed here)

One-against-rest

- Assume there are k possible target classes $(y_1, \ldots, y_k)$
- Train a classifier for each target class y_i $(i \in [1..k])$
    - Instances that belong to y_i are positive examples
    - All other instances y_j, $j \neq i$ are negative examples
- Combining predictions
    - If an instance is classified positive, the positive class gets a vote
    - If an instance is classified negative, all classes except for the positive class receive a vote

Example

- 4 classes (y_1, y_2, y_3, y_4)
- Classifying a given test instance (dots indicate the votes cast):

y_1	$+$	●
y_2	-	
y_3	-	
y_4	-	
Pred.	$+$	

y_1	-	●
y_2	$+$	
y_3	-	●
y_4	-	●
Pred.	-	

y_1	-	●
y_2	-	●
y_3	$+$	
y_4	-	●
Pred.	-	

y_1	-	●
y_2	-	●
y_3	-	●
y_4	$+$	
Pred.	-	

- Sum votes received: $(y_1$,●●●●), $(y_2$,●●), $(y_3$,●●), $(y_4$,●●)

One-against-one

- Assume there are k possible target classes $(y_1, \ldots, y_k)$
- Construct a binary classifier for each pair of classes (y_i, y_j)
  - $\frac{k \cdot (k-1)}{2}$ binary classifiers in total
- Combining predictions
  - The predicted class receives a vote in each pairwise comparison

Example

- 4 classes (y_1, y_2, y_3, y_4)
- Classifying a given test instance (dots indicate the votes cast):

y_1	+	•
y_2	-	
Pred.	+	

y_1	+	•
y_3	-	
Pred.	+	

y_1	+	
y_4	-	•
Pred.	-	

y_2	+	•
y_3	-	
Pred.	+	

y_2	+	
y_4	-	•
Pred.	-	

y_3	+	•
y_4	-	
Pred.	+	

- Sum votes received: $(y_1$,••), $(y_2$,•), $(y_3$,•), $(y_4$,••)

Evaluating multiclass classification

- Accuracy can still be computed as

$$ACC = \frac{\text{\#correctly classified instances}}{\text{\#total number of instances}}$$

- For other metrics
  - View it as a set of k binary classification problems (k is the number of classes)
  - Create confusion matrix for each class by evaluating "one against the rest"
  - Average over all classes

Confusion matrix

		Predicted				
		1	2	3	...	k
Actual	1	**24**	0	2		0
	2	0	**10**	1		1
	3	1	0	**9**		0
	...					
	k	2	0	1		**30**

Binary confusion matrices, one-against-rest

		Predicted				
		1	2	3	...	k
Actual	1	**24**	0	2		0
	2	0	**10**	1		1
	3	1	0	**9**		0
	...					
	k	2	0	1		**30**

$\Rightarrow$

		Predicted	
		1	$\neg$1
Act.	1	TP=24	FN=3
	$\neg$1	FP=2	TN=52

		Predicted	
		2	$\neg$2
Act.	2	TP=10	FN=2
	$\neg$2	FP=0	TN=69

...

For the sake of this illustration, we assume that the cells which are not shown are all zeros.

Averaging over classes

- Averaging can be performed on the instance level or on the class level
- **Micro-averaging** aggregates the results of individual instances across all classes
  - All instances are treated equal
- **Macro-averaging** computes the measure independently for each class and then take the average
  - All classes are treated equal

Micro-averaging

- **Precision**

$$P_\mu = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k}(TP_i + FP_i)}$$

- **Recall**

$$R_\mu = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k}(TP_i + FN_i)}$$

- **F1-score**

$$F1_\mu = \frac{2 \cdot P_\mu \cdot R_\mu}{P_\mu + R_\mu}$$

		predicted	
		i	$\neg i$
actual	i	TP_i	FN_i
	$\neg i$	FP_i	TN_i

Macro-averaging

- **Precision**

$$P_M = \frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i + FP_i}}{k}$$

- **Recall**

$$R_M = \frac{\sum_{i=1}^{k} \frac{TP_i}{TP_i + FN_i}}{k}$$

- **F1-score**

$$F1_M = \frac{\sum_{i=1}^{k} \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}}{k}$$

  - where P_i and R_i are Precision and Recall, respectively, for class i

		predicted	
		i	$\neg i$
actual	i	TP_i	FN_i
	$\neg i$	FP_i	TN_i

Text classification evaluation

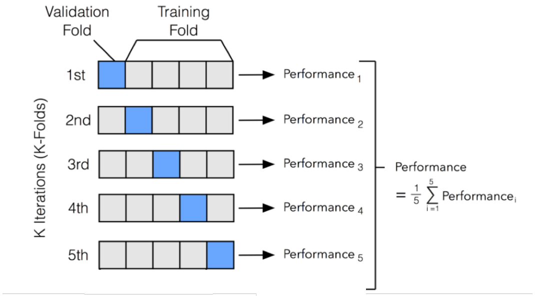
- Evaluating binary classification
- Evaluating multiclass classification
- Model development

Using a validation set

- Idea: hold out part of the training data for testing into a **validation set**
- **Single train/validation split**
  - Split the training data into $X\%$ training split and $100 - X\%$ validation split (an 80/20 split is common)

Using a validation set[2]

- k**-fold cross-validation**
  - Partition the training data randomly into k folds
  - Use $k-1$ folds for training and test on the kth fold; repeat k times (each fold is used for testing exactly once)
  - k is typically 5 or 10
  - Extreme: k is the number of data points, to maximize the number of training material available (called "leave-one-out" evaluation)



[2] Image source:
http://ethen8181.github.io/machine-learning/model_selection/model_selection.html

Summary

- Problem of text classification (binary and multiclass variants)
- Feature-bases text classifiers (bag-of-words representation, document-term matrix)
- Evaluation (confusion matrix, binary/multiclass)
- Evaluation measures (accuracy, precision, recall, F1, micro- and macro-averaging)
- Training/test splits, cross-validation

Reading

- Text Data Management and Analysis (Zhai&Massung)
    - Chapter 15 (Sections 15.1–15.4, 15.5.2, 15.6)