Introduction to Classification

LING 570

Fei Xia

Week 8: 11/16/09

Outline

- What is a classification problem?
- How to solve a classification problem?
- Case study

What is a classification problem?

An example: text classification task

- Task: given an article, predict its category.

- Categories:
  - Politics, sports, entertainment, travel, …
  - Spam or not spam

- What kind of information is useful to solve the problem?

Classification task

- Task:
  - C is a finite set of labels (a.k.a. categories, classes)
  - Given a x, decide its category $y \in C$.
- Instance: (x, y)
  - x: the thing to be labeled/classified
  - $y \in C$.
- Data: a set of instances
  - Labeled data: y is known
  - Unlabeled data: y is unknown
- Training data, test data

More examples

- Spam filtering
- Call center
- Sentiment detection
  - Good vs. Bad
  - 5-star system: 1, 2, 3, 4, 5

POS tagging

- Task: given a sentence, predict the tag of each word in the sentence.
- Is it a classification problem?
- Categories: noun, verb, adjective, …
- What information is useful?
- What are the differences between the text classification task and POS tagging?

  ➔ Sequence labeling problem

Tokenization / Word segmentation

- Task: given a string, break it into words.

- Categories:
  - NB (no break), B (with break)
  - B (beginning), I (inside), E (end)

- Ex: c1 c2 || c3 c4 c5
  - c1/NB c2/B c3/NB c4/NB c5/B
  - c1/B c2/E c3/B c4/I c5/E

- Relation to POS tagging?

How to solve
a classification problem?

Two stages

- Training stage
  - **Learner**: Training data ➔ classifier

- Testing stage
  - **Decoder**: Test data * classifier ➔ classification results

- Others:
  - Preprocessing stage
  - Postprocessing stage
  - Evaluation

How to represent x?

- The number of possible values for x could be infinite.

- Representing x as a feature vector:

  $x=<v_1,v_2,\ldots, v_n>$

  $x=<f_1=v_1,f_2=v_2,\ldots, f_n=v_n>$

- What is a good feature?

An example

Task: text classification

Categories: sports, entertainment, living, politics, …

doc1 debate immigration Iraq …

doc2   suspension Dolphins receiver …

doc3 song filmmakers charts rap ….

Training data: attribute-value table (Input to the training stage)

	f_1	f_2	…	f_K	Target
x_1	0	1	2.5	-1000	c_2
x_2	2.5	0	0	20	c_1
x_3					
…					
x_n					

A classifier

- It is the output of the training stage.

- Narrow definition:
  - $f(x) = y$, x is input, $y \in C$

- More general definition:
  - $f(x) = \{(c_i, score_i)\}$, $c_i \in C$.

Test stage

- Input: test data and a classifier

- Output: a decision matrix.

	x_1	x_2	x_3	….
c_1	0.1	0.4	0	…
c_2	0.9	0	0	…
c_3	0	0.1	0.4	
c_4	0	0.5	0.6	

Evaluation

Gold System	+	-
+	TP	FP
-	FN	TN

- Precision = TP/(TP+FP)
- Recall = TP/(TP+FN)
- F-score = 2PR/(P+R)

- Accuracy=(TP+TN)/(TP+TN+FP+FN)

- F-score or Accuracy?
- Why F-score?

An Example

Gold System	+	-
+	1	4
-	5	90

- Accuracy=91%

- Precision = 1/5
- Recall = 1/6
- F-score = $\frac{2*1/5*1/6}{1/5+1/6} = 2/11$

Steps for solving a classification task

- Prepare the data
  - Convert the task into a classification problem (optional)
  - Split data into training/dev/test
  - Convert the data into attribute-value table
- Training
- Testing
- Postprocessing (optional): convert the label sequence to something else
- Evaluation

Important subtasks (for you)

- Convert the problem into a classification task

- Converting the data into attribute-value table
  - Define feature types
  - Feature selection
  - Convert an instance into a feature vector

- Select a classification algorithm

Classification algorithms

- Decision Tree (DT)
- K nearest neighbor (kNN)
- Naïve Bayes (NB)
- Maximum Entropy (MaxEnt)*
- Supporting vector machine (SVM)**
- Conditional random field (CRF)**
- …

➔ Will be covered in LING572

More about attribute-value table

Attribute-value table

	f_1	f_2	…	f_K	Target
x_1	0	1	2.5	-1000	c_2
x_2	2.5	0	0	20	c_1
x_3					
…					
x_n					

Binary features vs. real-valued features

- Some ML methods can use real-valued features, others cannot.

- Very often, we convert real-valued features into binary ones.
  - temp 69
  - Use one threshold: IsTempBelow60 0
  - Use multiple thresholds:
    - TempBelow0 0 TempBet0And50 0 TempBet51And80 1 TempAbove81 0

Feature templates vs. Features

- A feature template: CurWord

- Corresponding features
  - CurWord_Mary
  - CurWord_the
  - CurWord_book
  - CurWord_buy
  - …

- One feature template corresponds to many features

Feature templates vs features (cont)

curWord = book

can be seen as a shorthand of

curWord_the=0   curWord_a=0
curWord_Mary=0 ….. curWord_book=1
…

An example

Mary will come tomorrow

	w_{-1}	w_0	$w_{-1}\ w_0$	w_{+1}	**y**
x1	<s>	Mary	<s> Mary	will	PN
x2	Mary	will	Mary will	come	V
x3	will	come	will come	tomorrow	V

This can be seen as a shorthand of a much bigger table.

Attribute-value table

- It is a very sparse matrix.

- In practice, it is often represented in a dense format.
  - Ex: x1=<f1=0 f2=0 f3=1 f4=0 f5=1 f6=0>

    x1 f3=1 f5=1

    x1 f3 f5

Case study

Case study (I)

- The NE tagging task
  - Ex: John visited New York last Friday.

    ➔ [person John] visited [location New York] [time last Friday]

- Is it a classification problem?
  - John/person-S visited New/location-B York/location-E last/time-B Friday/time-E

- What is x? What is y?

Case study (II)

- Task: identify tables in a document
- What is x? What is y?
- What features are useful?

An example

Table 4: Performance on the development set (the span number in the gold standard is 447)

Features	System span num	Classification accuracy	Exact match prec	Exact match recall	Exact match fscore	Partial match prec	Partial match recall	Partial match fscore
Regex templates	269	N/A	68.40	41.16	**51.40**	99.26	59.73	74.58
F_1	130	81.50	68.46	19.91	30.85	97.69	28.41	44.02
F_2	405	93.28	58.27	52.80	**55.40**	95.56	86.58	90.85
F_1+F_3	180	80.26	61.67	24.83	35.40	81.11	32.66	46.57
F_1+F_2	420	94.42	63.09	59.28	61.13	93.81	88.14	90.88
F_2+F_3	339	92.68	75.81	57.49	65.39	93.21	70.69	80.40
F_2+F_4	456	96.91	80.92	82.55	**81.73**	93.64	95.53	94.57
$F_1+F_2+F_3$	370	93.39	75.14	62.20	68.05	93.51	77.40	84.70
$F_1+F_2+F_4$	444	97.00	84.68	84.11	84.40	95.95	95.30	95.62
$F_2+F_3+F_4$	431	97.79	86.77	83.67	**85.19**	97.68	94.18	95.90
$F_1+F_2+F_3+F_4$	431	98.00	90.02	86.80	**88.38**	97.22	93.74	95.44

Table 5: Performance on the test set (the span number in the gold standard is 843)

Features	System span num	Classification accuracy	Exact match prec	Exact match recall	Exact match fscore	Partial match prec	Partial match recall	Partial match fscore
Regex templates	587	N/A	74.95	52.19	**61.54**	98.64	68.68	80.98
F_2	719	92.45	57.02	48.64	52.50	94.02	80.19	86.56
F_2+F_4	849	95.66	75.50	76.04	75.77	93.76	94.42	94.09
$F_2+F_3+F_4$	831	95.95	77.14	76.04	76.58	95.19	93.83	94.50
$F_1+F_2+F_3+F_4$	830	96.83	82.29	81.02	**81.65**	96.51	95.02	95.76

However, when we ran the same algorithm on the IGT data, the accuracy was only 50.2%.[10] In contrast, a heuristic approach that predicts the language ID according to the language names occurring in the document yields an accuracy of 65.6%.

Because the language name associated with an IGT instance almost always appears somewhere in the document, we propose to treat the language ID task as a reference resolution problem, where IGT instances are the *mentions* and the language names appearing in the document are the *entities*. A language identifier simply needs to link the mentions to the entities, allowing us to apply any good resolution algorithms such as (Soon et al., 2001; Ng,

for ODIN's data: bootstrapping NLP tools (specifically taggers), and providing search over ODIN's data (as a kind of large-scale multi-lingual search).

3.1 IGT for bootstrapping NLP tools

Since the target line in IGT data does not come with annotations (e.g., POS tags), it is first necessary to enrich it. Once enriched, the data can be used as a bootstrap for tools such as taggers.

3.1.1 Enriching IGT

In a previous study (Xia and Lewis, 2007), we proposed a three-step process to enrich IGT data: (1) parse the English translation with an English parser

Case study (III)

- Task: Co-reference task
  - Ex: John called Mary on Monday. She was not at home. He left a message on her answer machine.

- What is x? What is y?

- What features are useful?

Summary

- Important concepts
  - Instance: (x,y)
  - Labeled vs. unlabeled data
  - Training data vs. test data
  - Training stage vs. test stage
  - Learner vs. decoder
  - Classifier
  - Accuracy vs. precision / recall / f-score

Summary (cont)

- Attribute-value table vs. decision matrix
- Feature vs. Feature template
- Binary features vs. real-valued features
- Number of features can be huge
- Representation of attribute-value table