Text Categorization

Jacques Savoy
Université de Neuchâtel

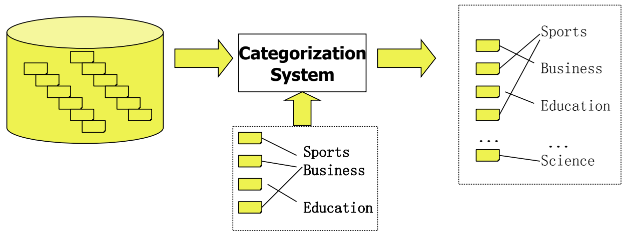
F. Sebastiani: Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1), 2002.
C.D. Manning & H. Schütze : *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge (MA), 2001.

Outline

- Examples & applications
- Definition of Text Categorization (TC)
- Rule-based and learning
- Dimension Reduction
- Text classifiers
- Evaluation

Example of TC

- Predefined categories C (categories may form hierarchy)
- Set of labeled document examples D (to learn)
- A standard classification (supervised learning) problem



Example of TC

- Instance language: <size, color, shape>
  - size ∈ {small, medium, large}
  - color ∈ {red, blue, green}
  - shape ∈ {square, circle, triangle}
- *C* = {positive, negative}
- *D*: (training & test) examples

Example	Size	Color	Shape	Category
1	small	red	circle	positive
2	large	red	circle	positive
3	small	red	triangle	negative
4	large	blue	circle	negative

Applications: Text Filtering

- Text Filtering
  - Classifying a stream of incoming documents (e.g., produced by a news agency for newspapers)
  - Usually single-label TC, splitting the new message into two disjoint categories {relevant, irrelevant} (e.g., e-mail into junk or ham)
  - May further classify relevant messages into various thematic categories (e.g., personalized web newspapers)
  - Text filtering may be installed at the producer end (selection based on user's profile)
  - Can be adapted from user feedback (adaptive filtering vs. routing or batch filtering)

5

Applications: Hierarchical Categorization

- Hierarchical categorization of Web pages
  - Large number of web pages useful to generate (automatically) a portal on a given topic (or generate an electronic catalogue)
  - Each category must have between $k_1 \leq x \leq k_2$ items
  - Must allow the creation of new categories (or to delete obsolete ones)
  - Can account for
    - Hypertextual nature of the document
    - Hierarchical nature of the categories (decomposing the classification into smaller classification problems)

6

Applications: Sentiment & Opinion

- Classifying web document (product review, customer information, social network) according to their opinionated content

- Fact
  "Five years ago, there were no Internet-related information businesses."
- Negative opinion
  "Since the United States is Korea's most important trade partner, the Korean economy was also affected immediately."
- Positive opinion
  "I believe that we have found the appropriate balance," he said.

7

Applications of TC

- Other applications
  - Document indexing
  - Word-Sense Disambiguation (WSD)
  - Multimedia document classification (through analysis of textual parts)
  - Author identification
  - Language identification
  - Text genre identification
  - Recommending messages / product
  - …

8

Problem Definition

- Need to assign a Boolean value {0,1} to each entry of the decision matrix
- C = {c_1,....., $c_{|C|}$} set of pre-defined categories, with |C|= *m*
- D = {d_1,..... d_n} set of documents to be categorized
- 1 for a_{ij}: d_j belongs to c_i (or True)
- 0 for a_{ij}: d_j does not belong to c_i (or False)

	d_1	...	...	d_j	...	...	d_n
c_1	a_{11}	...	...	a_{1j}	...	...	a_{1n}
...	...	...	...	...	...	...	...
c_i	a_{i1}	...	...	a_{ij}	...	...	a_{in}
...	...	...	...	...	...	...	...
c_m	a_{m1}	...	...	a_{mj}	...	...	a_{mn}

Problem Definition

- Categories are just symbolic labels (without additional knowledge about their meaning)
- No exogenous knowledge is available (based only on the docs without their metadata (type, author, source, etc.))
- Instead of a Boolean assignment, we may assign a probability (of belonging to the corresponding category)
- Given an integer *k*, exactly *k* (or ≤ *k*, or ≥ *k*) elements of C to be assigned to each d_j in D
- Single Label, *k* = 1, single label (non-overlapping)
  - Train a system which takes a d_i and C as input and outputs a c_i
- Multi-label, *k* in [0, |C|]
  - Train a system which takes a d_i and C as input and outputs C', a subset of C

Problem Definition

- Binary text classification
  - Each d_j in D must be assign either to c_i or to its complement
  - Build a separate system for each c_i, such that it takes in as input a d_i and outputs a Boolean value for (d_j, c_i)
  - The most general approach (multi-label into |C| binary classifier)
  - Based on assumption that decision on (d_j, c_i) is independent of (d_j, c_k)
- Binary text classification is more general
  - Many important applications
  - Solving the binary means solving the multi-label case
  - Many techniques are simply special case of the single-label case (and simpler to explain)

Problem Definition

- To choose a text classifier
  - Must generalize to classify correctly instances not in the training data
- Occam's razor
  - Prefer a simple hypothesis or rule agreeing with the data than a more complex one (and against the black box)
- Supervised or unsupervised
  - Supervised approaches need training examples

Steps in TC

1. Data processing
   - Term extraction,
     dimensionally reduction (Zipf's law, 50% of the words),
     feature selection
2. Define the test & training data
3. Creation of a classification model using the select algorithm
4. Model training (training set)
5. Model testing & evaluation (test set)
6. Final model building (using both training & test set)

Text Classifier

- Different strategies
  - Rule-based (expert system, Machine Learning)
  - Probabilistic classifier (**Naïve Bayes**)
  - Decision Tree classifier (see ML course)
  - Regression methods (see ML or stat course)
  - Neural Networks (see AI course)
  - Decision rule classifier
  - On-line methods
  - *tf.idf* method (see IR course)
  - Rocchio's method
  - Example-based classifiers (*k*-nearest-neighbor or k-NN)

Rule-Based Classifier

- Using an inductive rule learning, producing rules
  IF <*condition*> THEN <*category*>
- Condition: presence (or absence) of keyword in document descriptor (forming a Boolean condition)
  Decision: category assignment
- Example
  IF ((wheat & farm) or (wheat & community) or
  (bushels & export) or (wheat & tonnes) or
  (wheat & winter & ¬soft))
  THEN <WHEAT> ELSE ¬<WHEAT>
- Based on propositional logic
- Knowledge acquisition bottleneck

Rule-Based & Learning

- Use Machine Learning approaches
- Inductive process
- Given a set of documents classified (manually?) under category c_i, build a classifier by observing the underlying characteristics of documents belonging to category c_i or its complement (supervised learning)
- Must be able to classify unseen documents
- Pre-classified documents is the key resource
- Simple to classify documents than to extract rules
- Need to separate into two *disjoint* sets, the training set (to build the classifier and tune the parameters) and the test set (evaluation)

Document Representation

- Semantic is still a distant goal
- Need to build a compact text representation (indexing) with its meaningful units (lexical semantics)
  Assuming that compositional semantics is true
- Usually, we represent a document d_j by a vector of weighted term t_k (k=1, 2, …, t) (n-gram, isolated word, bigrams, noun phrase, …)

$$\left(w_{1j}, w_{2j}, \ldots, w_{kj} \ldots, w_{tj}\right)$$

  with $w_{kj} \geq 0$
- Give higher weight to most important terms

17

Document Representation

- Different ways to understand what is a term
- Usually based on bag-of-words
- Do not consider the location in the sentence
- May take account for the location of the sentence (e.g., title)
- Detecting phrases (syntactically, statistically) does not improve clearly the quality
- Can be a combined approach (isolated words, bigrams, noun phrases)

18

Document Representation

- Example
  1. Segmentation / tokenization
  2. Normalization (uppercase/lowercase, diacritics, punctuation, number, etc.)
  3. Stopword removal (the, in, of, with, has, done)
  4. Stemming (inflectional)

  Result: a bag-of-words
  Important step: need to weight each item in this bag.

19

Document Representation

- "The bill I'm signing today, known as the Weapons System Acquisition Reform Act, represents an important next step in this procurement reform process." (Obama, May, 22nd, 2009)
- "*the* bill i *m* signing today known *as the* weapons system acquisition reform act represents *an* important next step *in this* procurement reform process"
- "bill i sign*ing* today know*n* weapon*s* system acquisition reform act represent*s* important next step procurement reform process"
- "bill i sign today know weapon system acquisition reform act represent important next step procurement reform process"

20

Document Representation

- "Last night, Senator McCain said that George Bush won't be on the ballot this November." (Obama, October, 15nd, 2008)
- "last night senator mccain said *that* george bush *won t be on the* ballot *this* november"
- "last night senator mccain said george bush ballot november"
- "last night senator mccain said george bush ballot november"

21

Document Representation

- Indexing weights for term (feature) t_k in document D_i
  1. frequent terms must a have more weight: tf_{ik}
  2. words occurring in less documents (having a greater discrimination power) must have larger weight: $idf_k = \log(n/df_k)$ with n = # documents
  3. increase weights for smaller documents
  - the overall formula $w_{ik} = \mathrm{tf}_{ik} \cdot \mathrm{idf}_k$
  - many variations possible $w_{ik} = (\log(tf_{ik})+1) \cdot idf_k$

22

Term Selection

- Term selection by selecting terms receiving the higher scores according to a function
  - Using the document frequency (df_k)
  - Select terms having the highest df_k more valuable for TC (not for IR)
  - According to the Zipf's law, many terms have a low df
  - Example
    Removing term occurring in at less than x (training) documents (with x between 1 and 3)
  - Using both the tf_{jk} and idf_k values
    various other measures can be used (mutual information, χ^2, t-test, information gain, etc.)

23

Text Classifier

- By inductive learning
  - Define a CSV_i (Categorization Status Value) for each category c_i as:
  - CSV_i: $D \rightarrow$ {True, False} (hard classifier)
  - CSV_i: $D \rightarrow [0, 1]$ (ranking)
- We can apply thresholds
  - if $CSV_i(d_j) \geq \delta_i$ then assign c_i
  - May define different δ_i values for each category
  - These δ_i values an be learned
- Occam's rasor: Adopt the simplest hypothesis with equal performance (better generalization)

24

Bayes' Rule

- In the bar, a person said: "I win with a 7!"
  Does this person win when rolling a pair of dice or spinning a roulette? (our hypothesis H)
  Prob[dice | "7"], Prob[roulette | "7"]?
- Difficult to estimate directly…
- The prior: There is 6 tables, and in 2 they are playing with a roulette.
  - Prob[h_{dice}] = 4/6
  - Prob[$h_{roulette}$] = 2/6
- and the evidence (the "7")?

Bayes' Rule

- Evidence:
  - What is the chance to obtain a "7"?
- We need to compute the evidence
  (having a "7" according to the two hypothesis):
  Prob["7" | dice]? and Prob["7" | roulette]?
  - Prob["7" | dice] = Prob[e | h_{dice}] = 6/36
  - Prob["7" | roulette] = Prob[e | $h_{roulette}$] = 1/37
- Next we need to combine these two sources
  the prior and the likelihood (evidence)

Bayes' Rule

Thomas Bayes (1702-1761)

- Probability of event *H* given evidence *E:*

$$Prob[H|E] = \frac{Prob[E|H] \cdot Prob[H]}{Prob[E]}$$

- *A priori* probability of *H* : Prob[H]
  - Probability of event *before* evidence is seen
- *A posteriori* probability of *H* : Prob[H|E]
  - Probability of event *after* evidence is seen
- Combining prior probabilities and the likelihood of the data (according to the hypothesis *H*)

$$Prob[H|E] = \frac{Prob[E|H] \cdot Prob[H]}{Prob[E]} \propto Prob[E|H] \cdot Prob[H]$$

Bayes' Rule

- Prior:
  - Prob[h_{dice}] = 4/6
  - Prob[$h_{roulette}$] = 2/6
- Evidence:
  - Prob[e | h_{dice}] = 6/36
  - Prob[e | $h_{roulette}$] = 1/37
- Combination:

$$Prob[h_{dice}|e] = \frac{Prob[e|h_{dice}] \cdot Prob[h_{dice}]}{Prob[e]} = \frac{6/36 \cdot 4/6}{(6/36 \cdot 4/6) + (1/37 \cdot 2/6)}$$

$$Prob[h_{roulette}|e] = \frac{Prob[e|h_{roulette}] \cdot Prob[h_{roulette}]}{Prob[e]} = \frac{1/37 \cdot 2/6}{(6/36 \cdot 4/6) + (1/37 \cdot 2/6)}$$

Naïve Bayes Classifier

- In TC, we have
  - Evidence E = new document, sentence, instance
  - Event h_j = class value for this new instance
- The evidence can be divided into parts (i.e. the various features / terms E = {e_1, e_2, …e_n})
- Classify according to

$$h_{MAP} = \text{arg max}_{h_j \in H} \; Prob[h_j|e_1, e_2, \ldots e_n]$$

$$\begin{aligned} h_{MAP} &= \text{arg max}_{h_j \in H} \; \frac{Prob[e_1, e_2, \ldots e_n|h_j] \cdot Prob[h_j]}{Prob[e_1, e_2, \ldots e_n]} \\ &= \text{arg max}_{h_j \in H} \; Prob[e_1, e_2, \ldots e_n|h_j] \cdot Prob[h_j] \end{aligned}$$

29

Naïve Bayes Classifier

- The computation of $Prob[e_1, e_2, \ldots e_n|h_j]$ is in a general case too complex (interaction between the different e_i)

- The naïve Bayes classifier (conditionally independence)

$$Prob[e_1, e_2, \ldots e_n|h_j] \rightarrow \prod_{i=1}^{n} Prob[e_i|h_j]$$

and thus

$$h_{NB} = \text{arg max}_{h_j \in H} \; Prob[h_j] \cdot \prod_{i=1}^{n} Prob[e_i|h_j]$$

Naïve Bayes Classifier

- Hypotheses: {Spam, Ham} (binary decision)
- Evidence: an incoming email
  - The message is treated as a bag-of-words
- Knowledge
  - Prob[h_0=Spam] (with Prob[h_1=Ham] = 1 – Prob[h_0])
    - The prior probability of an e-mail message being a spam.
    - How to estimate this probability?
  - Prob[e_i|h_0=Spam]
    - the probability that a word is e_i if we know e_i is chosen from a spam.
    - How to estimate this probability?

31

Text Classification (Learning)

1. Collect all words, punctuation that occur in the C (Corpus)

   V ← the set of all distinct words or tokens (selection?, stemming?)
2. Compute the probability estimate P[h_j] and P[e_k|h_j] as

   doc_j ← the subset of documents from C having the target value is h_j

   P[h_j] = |doc_j| / |C| (reasonable prior estimation)

   $Text_j$ = concatenation of all members of doc_j

   n ← total number of words in $Text_j$

   for each word w_k in Voc

   n_k ← number of times word e_k occurs in $Text_j$

   P[e_k|h_j] = (n_k+1) / (n + |Voc|) (better than direct n_k / n)

   (smoothing the probabilities)

32

Example (Opinion Detection)

- Opinionated sentence (mixed)
- "Half of the job is psychiatry. "
  with "psychiatry" (*tf* = 1, *hapax*)
  NB: (0.179 / 0.821) half (3.23 / 5.91) job (2.61 / 2.13)
  psychiatry (-)
  → without opinion
- Opinionated sentence (negative)
- "You were often abused and humiliated "
  with "humiliated " (*tf*=1, *hapax*)
  NB: (0.397 / 0.603) you (12.65 / 7.7) often (4.17 / 3.39)
  abused (-)
  → without opinion

Evaluation

- Effectiveness measure on *unseen* examples (train & test)
- Contingency table for each category c_i
- TP: True positive
  TN: True negative
  FP: False positive
  FN: False negative

Category c_i		True state	
		Yes	No
Classifier decision	Yes	TP_i	FP_i
	No	FN_i	TN_i

- We can also create a global contingency table with all decisions (all documents)

Evaluation

- Precision (only the true) and Recall (all the truth)
- F_β measure (combining precision & recall)
  with $F_1 = (2 \cdot P \cdot R) / (P+R)$

$$Prec_i = \frac{TP_i}{TP_i + FP_i}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i}$$

$$F_\beta = \frac{(\beta^2 + 1) \cdot Prec_i \cdot Recall_i}{\beta^2 \cdot Prec_i \cdot Recall_i}$$

Conclusion

- TC is a major research area
- Many applications (proliferation of text-based information)
- Very useful when manual alternative is impossible
- Could be useful to help human taking the correct decision (suggesting possible solutions)
- A 100% correctness is impossible (humans are not consistent)
- In Naïve Bayes: independence between features
- Other challenges
  - Noisy text (OCR)
  - Speech transcripts
  - Multilingual TC
  - Other media (e.g., image categorization)

References

- C.D. Manning & H. Schütze : *Foundations of statistical natural language processing*. The MIT Press, Cambridge (MA), 2001.
- F. Sebastiani: Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1), 2002
- Y. Yang: An Evaluation of Statistical Approaches to Text Categorization. *Journal of Information Retrieval*, 1, 67-88, 1999.

37