Processing of large document collections

Part 4 (Information gain, boosting, text summarization)

Helena Ahonen-Myka

Spring 2005

In this part

- Term selection: information gain
- Boosting
- Text summarization

Term selection: information gain

- Information gain: measures the (number of bits of) information obtained for category prediction by knowing the presence or absence of a term in a document
- information gain is calculated for each term and the best n terms are selected

Term selection: IG

- information gain for term t:
  - m: the number of categories

$$G(t) = -\sum_{i=1}^{m} p(c_i)\log p(c_i)$$
$$+ p(t)\sum_{i=1}^{m} p(c_i \mid t)\log p(c_i \mid t)$$
$$+ p(\sim t)\sum_{i=1}^{m} p(c_i \mid \sim t)\log p(c_i \mid \sim t)$$

Example

- Doc 1: cat cat cat (c)
- Doc 2: cat cat cat dog (c)
- Doc 3: cat dog mouse (~c)
- Doc 4: cat cat cat dog dog dog (~c)
- Doc 5: mouse (~c)

- 2 classes: c and ~c

p(c) = 2/5, p(~c) = 3/5
p(cat) = 4/5, p(~cat) = 1/5, p(dog) = 3/5, p(~dog) = 2/5,
p(mouse) = 2/5, p(~mouse) = 3/5

p(c|cat) = 2/4, p(~c|cat) = 2/4, p(c|~cat) = 0, p(~c|~cat) = 1
p(c|dog) = 1/3, p(~c|dog) = 2/3, p(c|~dog) = 1/2, p(~c|~dog) = 1/2
p(c|mouse) = 0, p(~c|mouse) = 1, p(c|~mouse) = 2/3, p(~c|~mouse) = 1/3

-(p(c) log p(c) + p(~c) log p(~c)) = -(2/5 log 2/5 + 3/5 log 3/5)
= -(2/5 (log 2 – log 5) + 3/5 (log 3 – log 5)) = -(2/5 (1 – log 5) + 3/5 (log 3 – log 5))
= -(2/5 + 3/5 log 3 – log 5) = -(0.4 + 0.96 – 2.33) = 0.97 (log base = 2)

p(cat) (p(c|cat) log p(c|cat) + p(~c|cat) log p(~c|cat))
= 4/5 (1/2 log ½ + ½ log ½) = 4/5 log ½ = 4/5 (log 1 – log 2) = 4/5 (0 – 1) = -0.8

p(~cat) (p(c|~cat) log p(c|~cat) + p(~c|~cat) log p(~c|~cat))
= 1/5 (0 + 1 log 1) = 0

G(cat) = 0.97 – 0.8 – 0 = 0.17

p(dog) (p(c|dog) log p(c|dog) + p(~c|dog) log p(~c|dog))
= 3/5(1/3 log 1/3 + 2/3 log 2/3) = 3/5 (1/3 (log 1 – log 3) + 2/3 (log2 - log 3))
= 3/5 (-1/3 log 3 – 2/3 log 3 + 2/3) = 3/5(-log 3 + 2/3)
= 0.6 (-1.59 + 0.67) = -0.55

p(~dog) (p(c|~dog) log p(c|~dog) + p(~c|~dog) log p(~c|~dog))
= 2/5 (1/2 log ½ + ½ log ½) = 2/5 (log 1 – log 2) = -0.4

G(dog) = 0.97 – 0.55 – 0.4 = 0.02

p(mouse) (p(c|mouse) log p(c|mouse) + p(~c|mouse) log p(~c|mouse))
= 2/5 (0 + 1 log 1) = 0

p(~mouse) (p(c|~mouse) log p(c|~mouse) + p(~c|~mouse) log p(~c|~mouse))
= 3/5 (2/3 log 2/3 + 1/3 log 1/3) = -0.55

G(mouse) = 0.97 – 0 – 0.55 = 0.42

ranking: 1. mouse 2. cat 3. dog

7

Learners for text categorization: boosting

- the main idea of boosting:
  - combine many weak classifiers to produce a single highly effective classifier
- example of a weak classifier: "if the word 'money' appears in the document, then predict that the document belongs to category c"
  - this classifier will probably misclassify many documents, but a combination of many such classifiers can be very effective
- one boosting algorithm: AdaBoost

8

AdaBoost

- assume: a training set of pre-classified documents (as before)
- boosting algorithm calls a weak learner T times (T is a parameter)
  - each time the weak learner returns a classifier
  - error of the classifier is calculated using the training set
  - weights of training documents are adjusted
    - "hard" examples get more weight
  - the weak learner is called again
- finally the weak classifiers are combined

9

AdaBoost: algorithm

- Input:
  - N documents and labels: $<(d_1,y_1), \ldots,(d_N, y_N)>$, where $y_i \in \{-1, +1\}$
  - integer T: the number of iterations
- Initialize $D_1(i)$: $D_1(i) = 1/N$
- For s = 1,2,…,T do
  - Call WeakLearn and get a weak hypothesis h_s
  - Calculate the error of h_s: ε_s
  - Update the distribution (weights) of examples: $D_s(i) \rightarrow D_{s+1}(i)$
- Output the final hypothesis

10

Distribution of examples

- Initialize $D_1(i)$: $D_1(i) = 1/N$
- if N = 10 (there are 10 documents in the training set), the initial distribution of examples is:
  - $D_1(1) = 1/10$, $D_1(2) = 1/10$, …, $D_1(10) = 1/10$
- the distiribution describes the importance (=weight) of each example
- in the beginning all examples are equally important
  - later "hard" examples are given more weight

11

WeakLearn

- AdaBoost is a metalearner:
  - any learner could be used as a weak learner
  - typically very simple learners are used
  - a learner should be (slightly) better as random
    - error rate < 50%

12

WeakLearn

- idea: a classifier consists of one rule that tests the occurrence of one term
  - a document is in category c if and only if it contains this term
- to find the best term, the weak learner computes for each term the error

$$\varepsilon(t) = \sum_{i:t\in d_i, d_i\notin c} D_s(i) + \sum_{i:t\notin d_i, d_i\in c} D_s(i)$$

  - a good term discriminates between positive and negative examples
    - both occurrence and non-occurrence of a term can be significant

13

WeakLearn

- a term is chosen that minimizes $\varepsilon(t)$ or $1-\varepsilon(t)$
- let t_s be the chosen term
- the classifier h_s for a document d:

$$h_s(d) = \begin{cases} +1 & \text{if } t_s \in d \\ -1 & \text{if } t_s \notin d \end{cases}$$

14

Update weights

- the weights of training documents are updated
  - documents classified correctly get a lower weight
  - misclassified documents get a higher weight

$$D_{s+1}(i) = \frac{D_s(i)}{Z_s} \times \begin{cases} e^{-\alpha_s} & \text{if } h_s(d_i) = y_i \\ e^{\alpha_s} & \text{if } h_s(d_i) \neq y_i \end{cases}$$

15

Update weights

- calculate the error of h_s

$$\varepsilon_s = \sum_{i:h_s(d_i)\neq y_i} D_s(i)$$

- error = the sum of the weights of false positives and false negatives (in the training set)

16

Update weights

- calculation of α_s: (if error is small, α_s is large)

$$\alpha_s = \frac{1}{2}\ln\left(\frac{1-\varepsilon_s}{\varepsilon_s}\right)$$

- Z_s is a normalization factor
  - the weights have to form a distribution also after updates -> the sum of weights has to be 1

17

Final classifier

- the decisions of all weak classifiers are evaluated on the new document d and combined by voting:

$$h_{fin}(d) = \begin{cases} +1 & \text{if } \sum_{s=1}^{T} \alpha_s h_s(d) > 0 \\ -1 & \text{otherwise} \end{cases}$$

- note: α_s is also used to represent the goodness of the classifier s

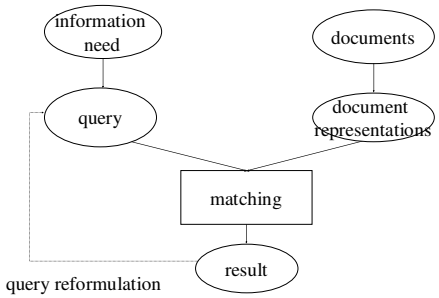
18

Performance of AdaBoost

- Schapire, Singer and Singhal (1998) have compared AdaBoost to Rocchio's method in text filtering
- experimental results:
  - AdaBoost is more effective, if a large number (hundreds) of documents are available for training
    - otherwise no noticeable difference
  - Rocchio is significantly faster

19

Mapping to the information retrieval process?



20

4. Text summarization

- "Process of distilling the most important information from a source to produce an abridged version for a particular user or task" (Mani, Maybury, 1999)

21

Text summarization

- many everyday uses:
  - news headlines (from around the world)
  - minutes (of a meeting)
  - tv digests
  - reviews (of books, movies)
  - abstracts of scientific articles
  - …

22

American National Standard for Writing Abstracts (1)

[Cremmins 82, 96]

- State the purpose, methods, results, and conclusions presented in the original document, either in that order or with an initial emphasis on results and conclusions.
- Make the abstract as informative as the nature of the document will permit, so that readers may decide, quickly and accurately, whether they need to read the entire document.
- Avoid including background information or citing the work of others in the abstract, unless the study is a replication or evaluation of their work.

23

American National Standard for Writing Abstracts (2)

[Cremmins 82, 96]

- Do not include information in the abstract that is not contained in the textual material being abstracted.
- Verify that all quantitative and qualitative information used in the abstract agrees with the information contained in the full text of the document.
- Use standard English and precise technical terms, and follow conventional grammar and punctuation rules.
- Give expanded versions of lesser known abbreviations and acronyms, and verbalize symbols that may be unfamiliar to readers of the abstract
- Omit needless words, phrases, and sentences.

24

Example

- Original version:

  There were significant positive associations between the concentrations of the substance administered and mortality in rats and mice of both sexes.

  There was no convincing evidence to indicate that endrin ingestion induced and of the different types of tumors which were found in the treated animals.

- Edited version:

  Mortality in rats and mice of both sexes was dose related.

  No treatment-related tumors were found in any of the animals.

25

Input for summarization

- a single document or multiple documents
- text, images, audio, video
- database

26

Characteristics of summaries

- extract or abstract
  - extract: created by reusing portions (usually sentences) of the input text verbatim
  - abstract: may reformulate the extracted content in new terms
- compression rate
  - ratio of summary length to source length
- connected text or fragmentary
  - extracts are often fragmentary

27

Characteristics of summaries

- generic or user-focused/domain-specific
  - generic summaries:

    summaries addressing a broad, unspecific user audience, without considering any usage requirements (general-purpose summary)
  - tailored summaries:

    summaries addressing group specific interests or even individualized usage requirements or content profiles (special-purpose summary)
    - expressed via query terms, interest profiles, feedback info, time window

28

Characteristics of summaries

- query-driven vs. text-driven summary
  - top-down: query-driven focus
    - criteria of interest encoded as search specifications
    - system uses specifications to filter or analyze relevant text portions.
  - bottom-up: text-driven focus
    - generic importance metrics encoded as strategies.
    - system applies strategies over representation of whole text.

Characteristics of summaries

- Indicative, informative, or critical summaries
  - indicative summaries

    summary has a reference function for selecting relevant documents for in-depth reading
  - informative summaries

    summary contains all the relevant (novel) information of the original document, thus substituting the original document
  - critical summaries

    summary not only contains all the relevant information but also includes opinions, critically assesses the quality of and the major assertions expressed in the original document

30

Architecture of a text summarization system

- Three phases:
  - analyzing the input text
  - transforming it into a summary representation
  - synthesizing an appropriate output form

31

The level of processing

- surface level
- discourse level

32

Surface-level approaches

- Tend to represent text fragments (e.g. sentences) in terms of shallow features
- the features are then selectively combined together to yield a salience function used to select some of the fragments

33

Surface level

- Shallow features of a text fragment
  - thematic features
    - presence of statistically salient terms, based on term frequency statistics
  - location
    - position in text, position in paragraph, section depth, particular sections
  - background
    - presence of terms from the title or headings in the text, or from the user's query

34

Surface level

- Cue words and phrases
  - "in summary", "our investigation"
  - emphasizers like "important", "in particular"
  - domain-specific bonus (+) and stigma (-) terms

35

Discourse-level approaches

- Model the global structure of the text and its relation to communicative goals
- structure can include:
  - format of the document (e.g. hypertext markup)
  - threads of topics as they are revealed in the text
  - rhetorical structure of the text, such as argumentation or narrative structure

36

Classical approaches

- Luhn '58
- general idea:
  - give a score to each sentence
  - choose the sentences with the highest score to be included in the summary

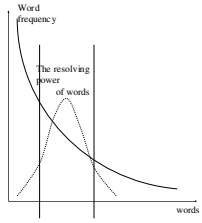
37

Luhn's method

- Filter terms in the document using a stoplist
- Terms are normalized based on combining together ortographically similar terms
  - differentiate, different, differently, difference
    -> differen
- Frequencies of combined terms are calculated and non-frequent terms are removed
- -> "significant" terms remain

38

Resolving power of words



- **Claim**: Important sentences contain words that occur "somewhat" frequently.
- **Method**: Increase sentence score for each frequent word.

[Luhn, 58]

39

Luhn's method

- Sentences are weighted using the resulting set of "significant" terms and a term density measure:
  - each sentence is divided into segments bracketed by significant terms not more than 4 non-significant terms apart
  - each segment is scored by taking the square of the number of bracketed significant terms divided by the total number of bracketed terms
    - score(segment) = significant_terms^2/all_terms

40

Exercise (CNN News)

- Let {13, computer, servers, Internet, traffic, attack, officials, said} be significant terms.
- "Nine of the 13 computer servers that manage global Internet traffic were crippled by a powerful electronic attack this week, officials said."

41

Exercise (CNN News)

- Let {13, computer, servers, Internet, traffic, attack, officials, said} be significant terms.
- * * * [13 computer servers * * * Internet traffic] * * * * * * [attack * * officials said]

42

Exercise (CNN News)

- [13 computer servers * * * Internet traffic]
  - score: 5^2 / 8 = 25/8 = 3.1
- [attack * * officials said]
  - score: 3^2 / 5 = 9/5 = 1.8

43

Luhn's method

- the score of the highest scoring segment is taken as the sentence score
- the highest scoring sentences are chosen to the summary
- a cutoff value is given, e.g.
  - N best terms, or
  - x% of the original text

44

"Modern" application

- text summarization of web pages on handheld devices (Buyukkokten, Garcia-Molina, Paepcke; 2001)
- macro-level summarization
- micro-level summarization

45

Web page summarization

- macro-level summarization
  - The web page is partitioned into 'Semantic Textual Units' (STUs)
    - Paragraphs, lists, alt texts (for images)
  - Hierarchy of STUs is identified
    - List - list item, table – table row
  - Nested STUs are hidden

46

Web page summarization

- micro-level summarization: 5 methods tested for displaying STUs in several states
  - incremental: 1) the first line, 2) the first three lines, 3) the whole STU
  - all: the whole STU in a single state
  - keywords: 1) important keywords, 2) the first three lines, 3) the whole STU

47

Web page summarization

  - summary: 1) the STUs 'most significant' sentence is displayed, 2) the whole STU
  - keyword/summary: 1) keywords, 2) the STUs 'most significant' sentence, 3) the whole STU

- The combination of keywords and a summary has given the best performance for discovery tasks on web pages

48

Web page summarization

- extracting summary sentences
  - Sentences are scored using a variant of Luhn's method:
    - Words are TF*IDF weighted; given a weight cutoff value, the high scoring words are selected to be significant terms
    - Weight of a segment: sum of the weights of significant words divided by the total number of words within a segment

49