Text Categorization (I)

Luo Si

Department of Computer Science
Purdue University
Text Categorization (I)

Outline

- Introduction to the task of text categorization
  - Manual v.s. automatic text categorization
- Text categorization applications
- Evaluation of text categorization
- K nearest neighbor text categorization method
Text Categorization

- **Tasks**
  - Assign predefined categories to text documents/objects

- **Motivation**
  - Provide an organizational view of the data

- **Large cost of manual text categorization**
  - Millions of dollars spent for manual categorization in companies, governments, public libraries, hospitals
  - Manual categorization is almost impossible for some large scale applications (Classification or Web pages)
Text Categorization

- Automatic text categorization
  - Learn algorithm to automatically assign predefined categories to text documents /objects
  - automatic or semi-automatic

- Procedures
  - **Training**: Given a set of categories and labeled document examples; learn a method to map a document to correct category (categories)
  - **Testing**: Predict the category (categories) of a new document

- Automatic or semi-automatic categorization can significantly reduce the manual efforts
Text Categorization: Examples

US Gamers Crack Puzzle in AIDS Research that Stumped Scientists for Years

In just three weeks, online gamers deciphered the structure of a retrovirus protein that has stumped scientists for over a decade, and a study out Sunday says their breakthrough opens doors for a new AIDS drug design.

Online gamers strike major blow in battle against AIDS

Video gamers solve microbiology puzzle

Highly Cited: Recommended: Gamers solve molecular puzzle that baffled

scientists

In Depth: Breakthrough For Retroviral Drug Design As Gamers Unfold Elusive Enzyme

Structure

Wikipedia: List of crowdsourcing projects

See all 70 sources »

Deep oceans store heat from global warming - for a while

The deep oceans can absorb enough heat to mask the effect of global warming for a decade at a time, say scientists at the National Center for Atmospheric Research (NCAR).
Text Categorization: Examples

Categories

- Computer Supported Cooperative Work (CSCW) (7)
- Conferences (7)
- Courses (2)
- Ergonomics@
- Information Architecture and Design@
- Institutes (19)
- Journals (1)
- Organizations (3)
- Projects (6)
- Web Directories (3)

SITE LISTINGS By Popularity | Alphabetical | (What's This?)

- HCI Bibliography
  Features abstracted validated bibliographic entries, along with a variety of reference materials.
  www.hcibib.org
Text Categorization: Examples

Medical Subject Headings (Categories)

1. Anatomy [A]
2. Organisms [B]
3. Diseases [C]
4. Chemicals and Drugs [D]
5. Analytical, Diagnostic and Therapeutic Techniques [E]
6. Psychiatry and Psychology [F]
   - Behavior and Behavior Mechanisms [F01]
   - Psychological Phenomena and Processes [F02]
   - Mental Disorders [F03]
   - Behavioral Disciplines and Activities [F04]
7. Biological Sciences [G]
Example: U.S. Census in 1990

- Included 22 million responses
- Needed to be classified into industry categories (200+) and occupation categories (500+)
- Would cost $15 millions if conducted by hand
- Two alternative automatic text categorization methods have been evaluated
  - Knowledge-Engineering (Expert System)
  - Machine Learning (K nearest neighbor method)
Example: U.S. Census in 1990

- **A Knowledge-Engineering Approach**
  - Expert System (Designed by domain expert)
  - Hand-Coded rule (e.g., if “Professor” and “Lecturer” -> “Education”)
  - Development cost: 2 experts, 8 years (192 Person-months)
  - Accuracy = 47%

- **A Machine Learning Approach**
  - K Nearest Neighbor (KNN) classification: details later; find your language by what language your neighbors speak
  - Fully automatic
  - Development cost: 4 Person-months
  - Accuracy = 60%
Many Applications!

- Web page classification (Yahoo-like category taxonomies)
- News article classification (more formal than most Web pages)
- Automatic email sorting (spam detection; into different folders)
- Word sense disambiguation (Java programming v.s. Java in Indonesia)
- Gene function classification (find the functions of a gene from the articles talking about the gene)
- What is your favorite applications?...
Techniques Explored in Text Categorization

- Rule-based Expert system (Hayes, 1990)
- **Nearest Neighbor methods** (Creecy’92; Yang’94)
- Decision symbolic rule induction (Apte’94)
- **Naïve Bayes** (Language Model) (Lewis’94; McCallum’98)
- Regression method (Furh’92; Yang’92)
- **Support Vector Machines** (Joachims 98, 05; Hofmann 03)
- Boosting or Bagging (Schapier’98)
- Neural networks (Wiener’95)
- ......
Text Categorization: Evaluation

Performance of different algorithms on Reuters-21578 corpus: 90 categories, 7769 Training docs, 3019 test docs, (Yang, JIR 1999)
### Contingency Table Per Category (for all docs)

<table>
<thead>
<tr>
<th></th>
<th>Truth: True</th>
<th>Truth: False</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted Positive</td>
<td>a</td>
<td>b</td>
<td>a+b</td>
</tr>
<tr>
<td>Predicted Negative</td>
<td>c</td>
<td>d</td>
<td>c+d</td>
</tr>
<tr>
<td></td>
<td>a+c</td>
<td>b+d</td>
<td>n=a+b+c+d</td>
</tr>
</tbody>
</table>

**a**: number of true positive docs  
**b**: number of false-positive docs  
**c**: number of false negative docs  
**d**: number of true-negative docs  
**n**: total number of test documents
Text Categorization: Evaluation

Contingency Table Per Category (for all docs)

n: total number of docs

Sensitivity: $a/(a+c)$  true-positive rate, the larger the better

Specificity: $d/(b+d)$  true-negative rate, the larger the better

They depend on decision threshold, trade off between the values
Recall: \( r = \frac{a}{a+c} \)  truly-positive (percentage of positive docs detected)

Precision: \( p = \frac{a}{a+b} \)  how accurate is the predicted positive docs

F-measure:

\[
F_\beta = \frac{(\beta^2 + 1)pr}{\beta^2 p + r} \quad F_1 = \frac{2pr}{p + r}
\]

Harmonic average:

\[
\frac{1}{\frac{1}{x_1} + \frac{1}{x_2}}
\]

Accuracy: \( \frac{a+d}{n} \)  how accurate is all the predicted docs

Error: \( \frac{b+c}{n} \)  error rate of predicted docs

Accuracy+Error=1
Text Categorization: Evaluation

- **Micro F1-Measure**
  - Calculate a single contingency table for all categories and calculate F1 measure
  - Treat each prediction with equal weight; better for algorithms that work well on large categories

- **Macro F1-Measure**
  - Calculate a single contingency table for every category; calculate F1 measure separately and average the values
  - Treat each category with equal weight; better for algorithms that work well on many small categories
K-Nearest Neighbor Classifier

- Also called “Instance-based learning” or “lazy learning”
  - low/no cost in “training”, high cost in online prediction
- Commonly used in pattern recognition (5 decades)
- Theoretical error bound analyzed by Duda & Hart (1957)
- Applied to text categorization in 1990’s
- Among top-performing text categorization methods
K-Nearest Neighbor Classifier

- Keep all training examples
- Find $k$ examples that are most similar to the new document ("neighbor" documents)
- Assign the category that is most common in these neighbor documents (neighbors vote for the category)
- Can be improved by considering the distance of a neighbor (A closer neighbor has more weight/influence)
K-Nearest Neighbor Classifier

- Idea: find your language by what language your neighbors speak

- Use K nearest neighbors to vote
  1-NN: Red; 5-NN: Blue; 10-NN: ?; Weight 10-NN: Blue
K Nearest Neighbor: Technical Elements

- Document representation
- Document distance measure: closer documents should have similar labels; neighbors speak the same language
- Number of nearest neighbors (value of K)
- Decision threshold
K Nearest Neighbor: Framework

Training data \( D = \{ (\vec{x}_i, y_i) \} \), \( \vec{x}_i \in \mathbb{R}^{|V|} \), docs, \( y_i \in \{0, 1\} \)

Test data \( \vec{x} \in \mathbb{R}^{|V|} \) The neighborhood is \( D_k \in D \)

Scoring Function \[
\hat{y}(\vec{x}) = \frac{1}{k} \sum_{\vec{x}_i \in D_k(\vec{x})} \text{sim}(\vec{x}, \vec{x}_i) y_i
\]

Classification:
\[
\begin{cases} 
1 & \text{if } \hat{y}(\vec{x}) - t > 0 \\
0 & \text{otherwise}
\end{cases}
\]

Document Representation: tf.idf weighting for each dimension
Choices of Similarity Functions

Euclidean distance
\[ d(x_1, x_2) = \sqrt{\sum_v (x_{1v} - x_{2v})^2} \]

Kullback Leibler distance
\[ d(x_1, x_2) = \sum_v x_{1v} \log \frac{x_{1v}}{x_{2v}} \]

Dot product
\[ x_1 \cdot x_2 = \sum_v x_{1v} x_{2v} \]

Cosine Similarity
\[ \cos(x_1, x_2) = \frac{\sum_v x_{1v} x_{2v}}{\sqrt{\sum_v x_{1v}^2} \sqrt{\sum_v x_{2v}^2}} \]

Kernel functions
\[ k(x_1, x_2) = e^{-d(x_1, x_2)/2\sigma^2} \] (Gaussian Kernel)

Automatic learning of the metrics
Choices of Number of Neighbors (K)

- Find desired number of neighbors by cross validation
  - Choose a subset of available data as training data, the rest as validation data
  - Find the desired number of neighbors on the validation data
  - The procedure can be repeated for different splits; find the consistent good number for the splits

When $n \to \infty$ (#docs), $k \to \infty$ (#neighbors) and $k/n \to 0$ (ratio of neighbors and total docs), KNN approaches minimum error.
Characteristics of KNN

Pros

- Simple and intuitive, based on local-continuity assumption
- Widely used and provide strong baseline in TC Evaluation
- No training needed, low training cost
- Easy to implement; can use standard IR techniques (e.g., tf.idf)

Cons

- Heuristic approach, no explicit objective function
- Difficult to determine the number of neighbors
- High online cost in testing; find nearest neighbors has high time complexity
Outline

- Introduction to the task of text categorization
  - Manual v.s. automatic text categorization
- Text categorization applications
- Evaluation of text categorization
- K nearest neighbor text categorization method
  - Lazy learning: no training
  - Local-continuity assumption: find your language by what language your neighbors speak
Bibliography

- D. D. Lewis. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. SIGIR, 1992