Enhanced hypertext categorization using hyperlinks

S. Chakrabarti, B. Dom, *IBM Almaden*

P. Indyk *Stanford University*

Proceedings of ACM SIGMOD Conference, 1998.

Outline

- **Introduction**
- **Text classification**
- **Hypertext classification**
  - **Radius-one specialization**
  - **Radius-two specialization**
- **Conclusion**

Introduction

- ❍ **Motivation**
  - ⌧ **an accurate classifier is an essential component of a hypertext database**
  - ⌧ **naive use of terms in the link neighborhood of a document can even degrade accuracy**
- ❍ **Goal**
  - ⌧ **a better classifier based on link information in a small neighborhood around documents**
  - ⌧ **adapt gracefully to the fraction of neighboring documents having known topics**

Introduction

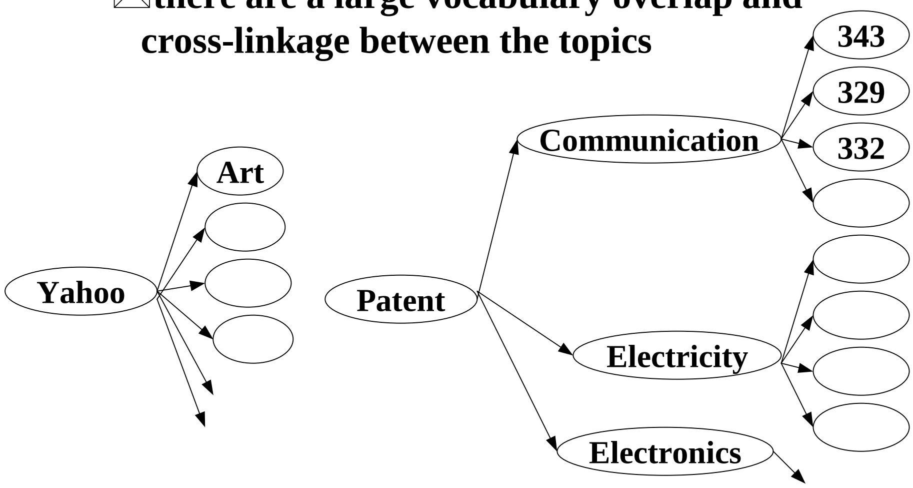
- ❍ **Problem**
  - ⌧ **diverse authorship**
  - ⌧ **navigational and citation links**
  - ⌧ **short, fragmented documents**
- ❍ **Challenge**
  - ⌧ **homogeneous corpora (IR)**
    - *TREC, Reuters, MEDLINE*
    - *correct rate: 80~87%*
  - ⌧ **hyperlinked corpora**
    - *US Patent Database: 64%, Yahoo!: 32%*

Introduction

- Obvious idea
  - include the text of a document's neighbors
    - *worse than the case based on only local text*
    - *link information is noisy*
- Main idea
  - the topics of neighboring documents determine linking behavior
  - initially guess the topics based on text alone, then update them iteratively
    - *Error rate: 21%*

Text classification

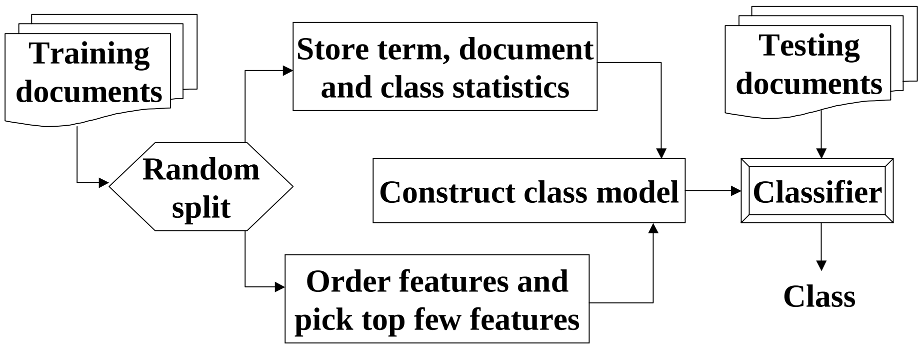
- Dataset
  - there are a large vocabulary overlap and cross-linkage between the topics



Text classification

❍**TAPER**

- **a basic classification engine**
  - *construct a diverse set of classifiers specifically suited to each internal node*



Text classification

❍**Feature selection**

- **good discriminators vs. noise**
- **order the term by decreasing ability to separate the classes**
- **formula:** $$score(t) = \frac{\sum_{c_1, c_2} \left(\mu(c_1, t) - \mu(c_2, t)\right)^2}{\sum_c \frac{1}{|c|} \sum_{d \in c} \left(f(t, d, c) - \mu(c, t)\right)^2}$$

❍**Class model**

- **Bernoulli model vs. binary model**
- **formula:**

$$\Pr[d \in c \mid c_0, F] = \frac{\pi(c) \prod_{t \in d \cap F} \theta(c, t)^{n(d,t)}}{\sum_{c' \in child(c_0)} \pi(c') \prod_{t \in d \cap F} \theta(c', t)^{n(d,t)}}$$

Text classification

- **Example**
  - **c_1: $<d_1, d_2>$, c_2: $<d_3, d_4>$**

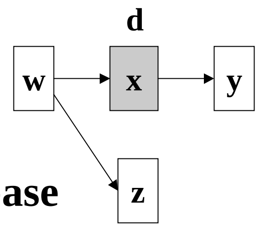
d_1: $<t_1{:}1, t_2{:}2, t_3{:}1>$, d_2: $<t_1{:}3, t_2{:}0, t_3{:}5>$
d_3: $<t_1{:}2, t_2{:}10, t_3{:}2>$, d_4: $<t_1{:}4, t_2{:}12, t_3{:}6>$
$\mu(c_1, t_1)=2, \mu(c_1, t_2)=1, \mu(c_1, t_3)=3$
$\mu(c_2, t_1)=3, \mu(c_2, t_2)=11, \mu(c_2, t_3)=4$

  - *$score(t_1)=1/2$, $score(t_2)=50$, $score(t_3)=1/8$*
- **$d' \cap F$: $<t_1{:}1, t_2{:}3>$**

$\theta(c_1,t_1)=1, \theta(c_1,t_2)=1/2, \theta(c_2,t_1)=1, \theta(c_2,t_2)=1$

  - *$Pr[d' \in c_1]=1/9$, $Pr[d' \in c_2]=8/9$*

Hypertext classification



- **Feature engineering**
  - **I: in-link, O: out-link**
  - **d: <x, O©y, I©w, IO©z >**
- **Experiment: US Patent Database**
  - **all immediate neighbors**
    - *Local: 36%*
    - *Local+Nbr: 38.3%*
    - *Local+TagNbr: 38.2%*
  - **term distribution is not sufficiently similar to the true class**

Hypertext classification

- ❍**Radius-one specification**
  - ⌧**if classes for all neighboring documents are known, replace each hyperlink with class ID**
  - ⌧**choose c_i to maximize $Pr(c_i|N_i)$**
  - ⌧**formula:**

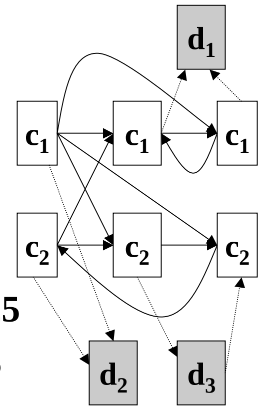
$$\Pr(N_i \mid c_i)\Pr(c_i) =$$

$$\pi(c_i)\prod_{j=1}^{m}\left[\phi(\gamma_j, c_i \mid I)\right]^{n(\gamma_j, i|I)}\left[\phi(\gamma_j, c_i \mid O)\right]^{n(\gamma_j, i|O)}$$

Hypertext classification

- ❍**Example**

$\phi(\gamma_1, c_1|I)=4/5$, $\phi(\gamma_2, c_1|I)=1/5$
$\phi(\gamma_1, c_1|O)=4/6$, $\phi(\gamma_2, c_1|O)=2/6$
$\phi(\gamma_1, c_2|I)=2/5$, $\phi(\gamma_2, c_2|I)=3/5$
$\phi(\gamma_1, c_2|O)=1/4$, $\phi(\gamma_2, c_2|O)=3/4$



d_1: $Pr(N_1| c_1)=16/25$, $Pr(N_2| c_2)=4/25$

d_2: $Pr(N_1| c_1)=4/25$, $Pr(N_2| c_2)=6/25$

d_3: $Pr(N_1| c_1)=1/15$, $Pr(N_2| c_2)=9/20$

- ⌧**$Pr(d_1\in c_1)=16/20$, $Pr(d_2\in c_2)=6/10$, $Pr(d_3\in c_2)=27/31$**

Hypertext classification

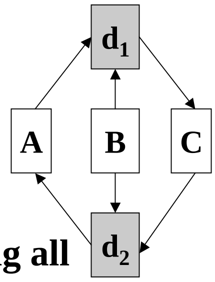
- ❍**Iterative relaxation labeling**
  - ⌧**if some or all of the neighboring classes are unknown**
    - *given test document d*
    - *construct a radius-r graph G(d) around d*
    - *assign initial classes to all $d_i \in G(d)$ using local text*
    - *iterate until consistent*
      - recompute the class for each $d_i \in G(d)$ based on local text and class of neighbors

Hypertext classification

- ❍**Experiment: US Patent Database**
  - ⌧**complete supervised case**
    - *Text: 36%*
    - *Link: 34%*
    - *Prefix: 22.1%*
    - *Text+Prefix: 21%*
  - ⌧**partially supervised case**
    - *Text: constant*
    - *Link: 34~31~27~24~22.1%*
    - *Text+Link: 26~25~24~22~21%*

Hypertext classification

- ❍**Radius-two specification**
  - ☒**pages that cites or are cited by many common pages are regarded as similar**
  - ☒**these common pages are called bridge**
    - *B is an IO-bridge for d_1 and d_2*
    - *A: II-bridge, C: OO-bridge*
    - *OI-bridge is not meaningful*
- ❍**Experiment: Yahoo!**
  - ☒**the fraction of coherent pairs among all pairs (d_1, d_2) where $(d1\text{-}d2)_B = D_j$, for some j**



Hypertext classification

- ❍**TAPER with IO-bridge**
  - ☒**assumed pure bridge**
  - ☒**take all prefixes of the known classes from pages that are IO-bridged to a training page**
- ❍**IO-bridge with locality**
  - ☒**include class ID c as a feature of page d, if**
    - *a bridge contains out-links to d_1, d, and d_2*
    - *the classes of d_1 and d_2 are the same (c)*
    - *no out-links between d_1 and d_2 point to a page with a known class*

Hypertext classification

- **Experiment: Yahoo!**
  - **error rate**
    - *Text: 68%*
    - *IO-bridge: 25%*
    - *IO-locality: 21%*
  - **coverage**
    - *Text: 100%*
    - *IO-bridge: 75%*
    - *IO-locality: 62%*

Conclusion

- **Contribution**
  - **This is the first topic classification system that combines textual and linkage features**
  - **achieve significantly improved accuracy at a moderate computational overhead**
    - *US Patent Database: 36~21%*
    - *Yahoo!: 68~21%*
- **Comment**
  - **the classifier only needs very few features**
  - **management of link information is required**