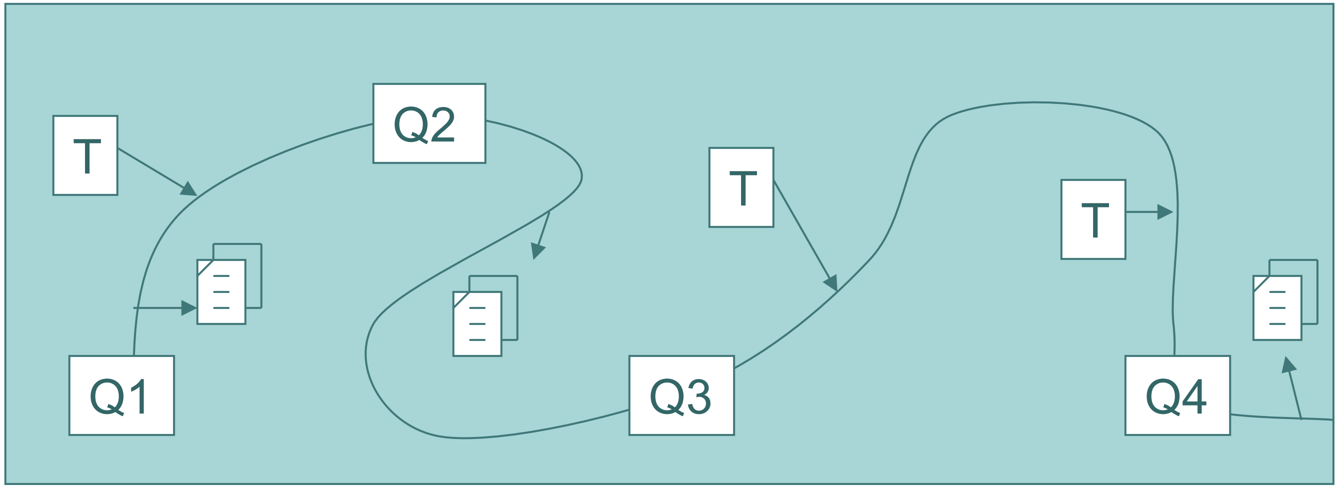
Digital Libraries

Collaborative Filtering and Recommender Systems

Min-Yen KAN

Information Seeking, recap



In information seeking, we may seek others' opinion:

- Recommender systems may use collaborative filtering algorithms to generate their recommendations

What is its relationship to IR and related fields?

Is it IR? Clustering?

- Information Retrieval:
  - Uses content of document

- Recommendation Systems:
  - Uses item’s metadata

    Item – item recommendation
  - Collaborative Filtering

    User – user recommendation
    1. Find similar users to current user,
    2. Then return their recommendations

  Clustering can be used to find recommendations

Collaborative Filtering

- Effective when untainted data is available
- Typically have to deal with sparse data
  - Users will only vote over a subset of all items they've seen

- Data:
  - Explicit: recommendations, reviews, **ratings**
  - Implicit: query, browser, past purchases, session logs

- Approaches
  - **Model based** – derive a user model and use for prediction
  - **Memory based** – use entire database

- Functions
  - Predict – predict ranking for an item
  - Recommend – produce *ordered* list of items of interest to the user.

Why are these two considered distinct?

Memory-based CF

- Assume active user *a* has ranked *I* items:
- Mean ranking given by:

$$\overline{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$$

A specific vote for an item j

- Expected ranking of a new item given by:

$$p_{a,j} = \overline{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \overline{v}_i)$$

Rating of past user

normalization factor

Correlation of past user with active one

Correlation

- How do find similar users?
  - Check correlation between active user’s ratings and yours
  - Use Pearson correlation:

$$w(a,i) = \frac{\sum_j (v_{a,j} - \overline{v}_a)(v_{i,j} - \overline{v}_i)}{\sqrt{\sum_j (v_{a,j} - \overline{v}_a)^2 \sum_j (v_{i,j} - \overline{v}_i)^2}}$$

    - Generates a value between 1 and -1
    - 1 (perfect agreement), 0 (random)

Similarity can also be done in terms of vector space.
What are some ways of applying this method to this problem?

Two modifications

- Sparse data
  - Default Voting
    - Users would agree on some items that they didn’t get a chance to rank
    - Assume all unobserved items have neutral or negative ranking.
    - Or impute values based on available data
    - Smoothes correlation values in sparse data

- Balancing Votes:
  - Inverse User Frequency
    - Universally liked items not important to correlation
    - Weight (j) = ln (# users/# users voting for item j)

Model-based methods: NB Clustering

Assume all users belong to several different types $C = \{C_1, C_2, \ldots, C_n\}$

- Find the model (class) of active user
  - Eg. Horror movie lovers
  - This class is hidden
- Then apply model to predict vote

$$\Pr(C = c, v_1, \ldots, v_n) = \Pr(C = c) \prod_{i=1}^{n} \Pr(v_i | C = c)$$

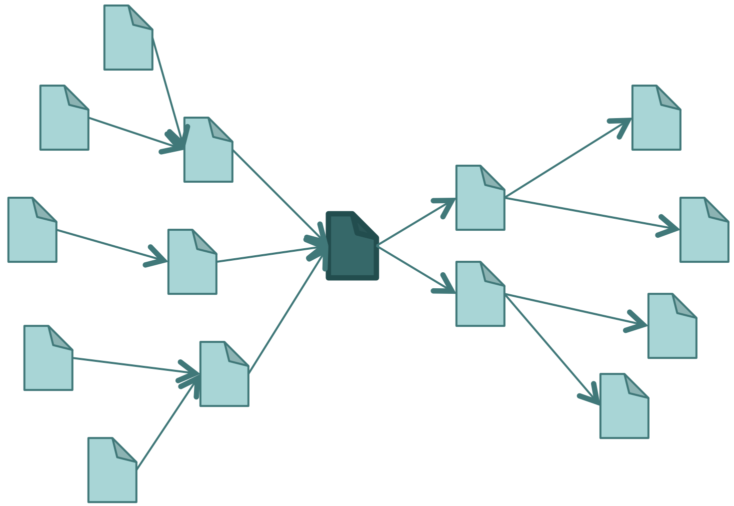
Class probability

Probability of a vote on item i given class C

Scholarly Paper Recommendation

- Leverage the citation network
  - Usage data also possible

- Use context to combat sparse data (Sugiyama and Kan, 2010; 2011)



Using the Citation Network

- Use the body of the paper, certain sections (e.g., references) or windows around in-text citations (citances).

- With respect to particular authors as users, can also use their publications as a user model
  - Needs to be weighted appropriately

Detecting untainted data

- *Shill* = a decoy who acts enthusiastically in order to stimulate the participation of others

- Push: cause an item's rating to rise
- Nuke: cause an item's rating to fall

Properties of shilling

Given current user-user recommender systems:

- An item with more variable recommendations is easier to shill
- An item with less recommendations is easier to shill
- An item farther away from the mean value is easier to shill towards the same direction

How would you attack a recommender system?

Attacking a recommender system

- Introduce new users who rate target item with high/low value

How do you make this shill less noticeable?

Shilling, continued

- Recommendation ***is*** different from prediction
  - Recommendation produces *ordered* list, most people only look at first *n* items

- Obtain recommendation of new items before releasing item
  - Default Value

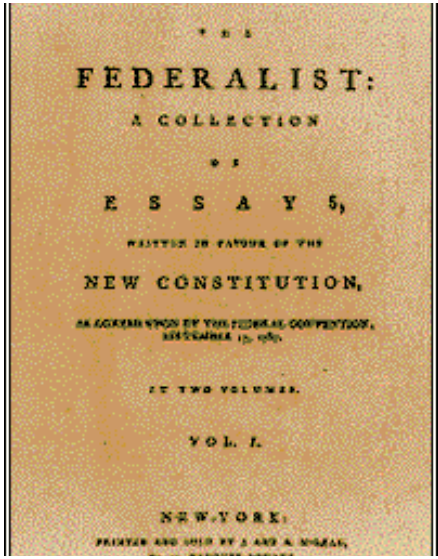
Digital Libraries

Computational Literary Analysis

Min-Yen KAN

The Federalist Papers

- A series of 85 papers written by Jay, Hamilton and Madison
- Intended to help persuade voters to ratify the US constitution
- Most of the papers have attribution but the authorship of 12 papers are disputed
  - Either Hamilton or Madison
- Want to determine who wrote these papers
  - Also known as textual forensics



Hamilton

Madison

Wordprint and Stylistics

- Claim: Authors leave a unique *wordprint* in the documents which they author
- Claim: Authors also exhibit certain *stylistic patterns* in their publications

Feature Selection

- Content-specific features (Foster 90)
  - key words, special characters
- Style markers
  - Word- or character-based features (Yule 38)
    - length of words, vocabulary richness
  - Synonym pairs (but very few)
  - Function words (Mosteller & Wallace 64)
- Structural features
  - Email: Title or signature, paragraph separators (de Vel *et al.* 01)
  - Can generalize to HTML tags
  - To think about: artifact of authoring software?

Bayes Theorem on function words

- M & W examined the frequency of 100 function words
- Smoothed these frequencies using negative binomial (not Poisson) distribution

Frequency	Hamilton	Madison
0	.607	.368
1	.303	.368
2	.0758	.184

- Used Bayes' theorem and linear regression to find weights to fit for observed data

- Sample words:

as	do	has	is	no	or	than	this
at	down	have	it	not	our	that	to
be	even	her	its	now	shall	the	up

A Funeral Elegy and *Primary Colors*

"Give anonymous offenders enough verbal rope and column inches, and they will hang themselves for you, every time" – Donald Foster in *Author Unknown*

- *A Funeral Elegy*: Foster attributed this poem to W.S.
  - Initially rejected, but identified his anonymous reviewer
  - But about a decade late (2002), he was "proven" wrong
- Forster also attributed *Primary Colors* to Newsweek columnist Joe Klein
- Analyzes text mainly by hand

Foster’ s features

- Very large feature space, look for distinguishing features:
  - Topic words
  - Punctuation
  - Misused common words
  - Irregular spelling and grammar

- Some specific features (most compound):
  - Adverbs ending with “y”: *talky*
  - Parenthetical connectives: … , *then*, …
  - Nouns ending with “mode”, “style”: *crisis mode*, *outdoor-stadium style*

Typology of English texts

- Biber (89) typed different genres of texts

- Five dimensions …
  1. Involved vs. informational production
  2. Narrative?
  3. Explicit vs. situation-dependent
  4. Persuasive?
  5. Abstract?

… targeting these genres
  1. Intimate, interpersonal interactions
  2. Face-to-face conversations
  3. Scientific exposition
  4. Imaginative narrative
  5. General narrative exposition

Features used (*e.g.*, Dimension 1)

- Biber also gives a feature inventory for each dimension

Features	
THAT deletion Contractions BE as main verb WH questions 1st person pronouns 2nd person pronouns General hedges	+
Nouns Word Length Prepositions Type/Token Ratio	–

Score	Genre
35	Face to face conversations
30	
25	
20	Personal Letters Interviews
15	
10	
5	
	Prepared speeches
0	
	General fiction
-5	
-10	Editorials
-15	Academic prose; Press reportage Official Documents
-20	

Discriminant analysis for text genres

- Karlgren and Cutting (94)
  - Same text genre categories as Biber
  - Simple count and average metrics
  - Discriminant analysis (in SPSS)
  - 64% precision over four categories

Some count features

- Adverb
- Character
- Long word (> 6 chars)
- Preposition
- 2nd person pronoun
- "Therefore"
- 1st person pronoun
- "Me"
- "I"
- Sentence

Other features

- Words per sentence
- Characters per word
- Characters per sentence
- Type / Token Ratio

Conclusions

- Marked vocabulary or syntax of limited use as it doesn’t occur often
- Top n words thrown through PCA provides a reasonable baseline for state-of-the art

Copy detection

Prevention –

stop or disable copying process

Detection –

decide if one source is the same as another

Copy / duplicate detection

- Compute signature for documents
  - Register signature of authority doc
  - Check a query doc against existing signature
- Variations:
  - Length: document / sentence* / window
  - Signature: checksum / keywords / phrases

Granularity

- Large chunks
  - Lower probability of match, higher threshold
- Small chunks
  - Smaller number of unique chunks
  - Lower search complexity

Subset problem

- If a document consists of just a subset of another document, standard VS model may show low similarity
  - Example: Cosine $(D_1,D_2) = .61$
    D_1: <A, B, C>,
    D_2: <A, B, C, D, E, F, G, H>

- Shivakumar and Garcia-Molina (95): use only *close* words in VSM
  - **Close** = comparable frequency, defined by a tunable ε distance.

R-measure

- Normalized sum of lengths of all suffixes of the text repeated in other documents

$$R^2(T \mid T_1, \ldots T_m) = \frac{2}{l(l+1)} \sum_{i=1}^{l} Q(T[i..l] \mid T_1, \ldots T_m),$$

where $Q(S|T_1 \ldots T_n)$ = length of longest prefix of S repeated in any one document

  - Computed easily using suffix array data structure
  - More effective than simple longest common substring

R-measure example

T = cat_sat_on

T1 = the_cat_on_a_mat

T2 = the_cat_sat

$$\frac{2}{l(l+1)}\sum_{i=1}^{l} Q(T[i..l] \mid T_1, \ldots T_m),$$

$$R^2(T|T_1,T_2) = \frac{2}{10 \times (10 + 1)} ((7+6+5+4+3) + (5+4+3+2+1))$$

cat_sat
at_sat
t_sat
_sat
sat

at_on
t_on
_on
on
n

Computer program plagiarism

- Use stylistic rules to compile fingerprint:
  - Commenting
  - Variable names
  - Formatting
  - Style (e.g., K&R)

- Use this along with program structure
  - Edit distance
  - To think about: What about hypertext structure?

```
/************************************
* This function concatenates the first and
* second string into the third string.
*************************************
void strcat(char *string1, char *string2, char
     *string3)
{
 char *ptr1, *ptr2;
 ptr2 = string3;
/*
 * Copy first string
 */
for(ptr1=string1;*ptr1;ptr1++) {
*(ptr2++) = *ptr1;
}
```

```
/*
 * concatenate s2 to s1 into s3.
 * Enough memory for s3 must already be
     allocated. No checks !!!!!!
 */
mysc(s1, s2, s3)
      char *s1, *s2, *s3;
{
  while (*s1)
    *s3++ = *s1++;

  while (*s2)
    *s3++ = *s2++;
}
```

Conclusion

- Find attributes that are stable between (low variance) texts for a collection, but differ across different collections
- Difficult to scale up to many authors and many sources
  - Most work only does pairwise comparison
  - Clustering may help as a first pass for plagiarism detection

To think about…

- The Mosteller-Wallace method examines function words while Foster's method uses key words. What are the advantages and disadvantages of these two different methods?
- What are the implications of an application that would emulate the wordprint of another author?
- What are some of the potential effects of being able to undo anonymity?
- Self-plagiarism is common in the scientific community. Should we condone this practice?