Probabilistic Latent Semantic Analysis

Shuguang Wang

Advanced ML

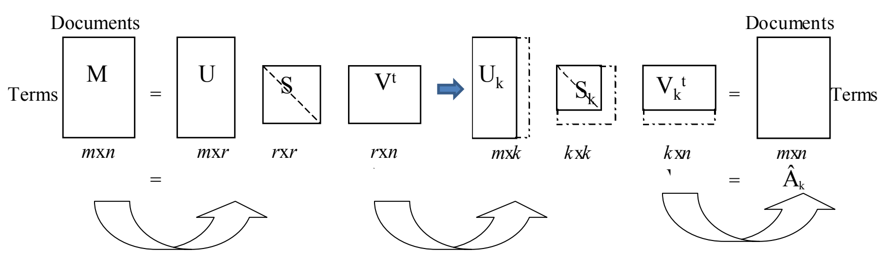
Outline

- Review Latent Semantic Indexing/Analysis (LSI/LSA)
  - LSA is a technique of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms.
  - In the context of its application to information retrieval, it is called LSI.
- Probabilistic Latent Semantic Indexing/Analysis (PLSI/PLSA)
- Hypertext-Induced Topic Selection (HITS and PHITS)
- Joint model of PHITS and PLSI

Review: Latent Semantic Analysis/Indexing

- Perform a low-rank approximation of document-term matrix
- General idea
  - Assumes that there is some underlying or *latent* structure in word usage that is obscured by variability in word choice
  - Instead of representing documents and queries as vectors in a t-dimensional space of terms, represent them (and terms themselves) as vectors in a lower-dimensional space whose axes are concepts that effectively group together similar words
  - These axes are the Principal Components from PCA
  - Compute document similarity based on the inner product in the latent semantic space (cosine metric)

Review: LSI Process

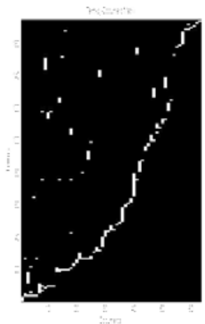


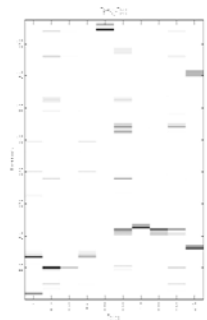
SVD:
Convert term-by-document matrix into 3matrices U, S and V

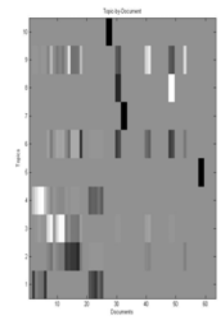
Dimension Reduction:
Ignore zero and low-order rows and columns

Reconstruct Matrix:
Use the new matrix to process queries
OR, map query to reduced space

Review: LSI Example







Term by Document Matrix (174 x 63) → SVD → Term by Topic Matrix (174 x 10) | Topic by Document Matrix (10x 63)

U | V^T

Review: LSA Summary

- Pros:
  - Low-dimensional document representation is able to capture synonyms. Synonyms will fall into same/similar concepts.
  - Noise removal and robustness by dimension reduction.
  - Exploitation of redundant data
  - Correlation analysis and Query expansion (with related words)
  - Empirical study shows it outperforms naïve vector space model
  - Language independent
  - high recall: query and document terms may be disjoint
  - Unsupervised/completely automatic

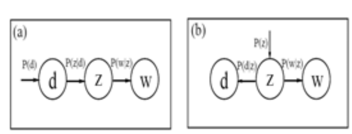
Review: LSA Summary

- Cons:
  - No probabilistic model of term occurrences.
  - Problem of polysemy (multiple meanings for the same word) is not addressed.
  - Implicit Gaussian assumption, but term occurrence is not normally distributed.
  - Euclidean distance is inappropriate as a distance metric for count vectors (reconstruction may contain negative entries).
  - Directions are hard to interpret.
  - Computational complexity is high: $O(\min(mn^2, nm^2))$ for SVD, and it needs to be updated as new documents are found/updated
  - ad hoc selection of the number of dimensions, model selection

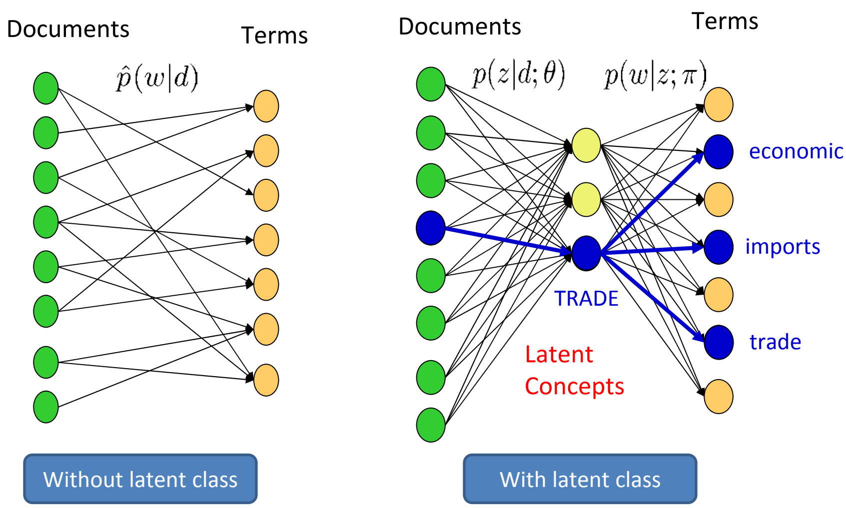
Probabilistic LSA: a statistical view of LSA

- Aspect Model
  - For co-occurrence data which associated with a latent class variable.
  - *d* and *w* are independent conditioned on *z, where d is document, w is term, z is concept*

$$
\begin{aligned}
P(d,w) = P(d)P(w \mid d) &= P(d)\sum_{z \in Z} P(w \mid z)P(z \mid d) \\
&= \sum_{z \in Z} P(d)P(w \mid z)P(z \mid d) \\
&= \sum_{z \in Z} P(d,z)P(w \mid z) \\
&= \sum_{z \in Z} P(z)P(w \mid z)P(d \mid z)
\end{aligned}
$$



PLSA Illustration



Why Latent Concept?

- Sparseness problem, terms not occurring in a document get zero probability
- “Unmixing” of superimposed concepts
- No prior knowledge about concepts required
- Probabilistic dimension reduction

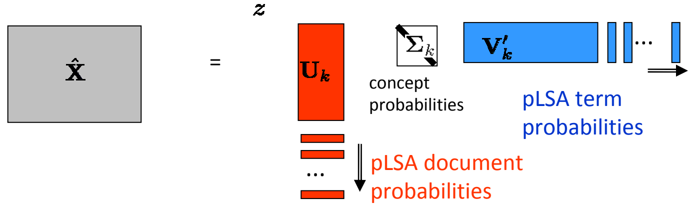
Quick Detour: PPCA vs. PLSA

- PPCA is also a probabilistic model.
- PPCA assume normal distribution, which is often not valid.
- PLSA models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions.
- Multinomial distribution is a better alternative in this domain.

PLSA Mixture Decomposition Vs. LSA/SVD

- PLSA is based on mixture decomposition derived from latent class model.

$$\hat{p}_{\mathrm{LSA}}(d,w) = \sum_{z} p(d|z)\, p(z)\, p(w|z)$$



- Different from LSA/SVD: **non-negative** and **normalized**

KL Projection

- Log Likelihood

$$L = \sum_{d \in D, w \in W} n(d,w) \log P(d,w)$$

$$\mathcal{L} = \sum_{d \in \mathcal{D}} n(d) \left[\underline{\sum_{w \in \mathcal{W}} \frac{n(d,w)}{n(d)} \log P(w|d)} + \log P(d) \right]$$

Recall KL divergence is $D_{\mathrm{KL}}(P \| Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$

$P = \hat{P}(w \mid d) = \frac{n(d,w)}{n(d)} \quad Q = P(w \mid d)$

Rewrite the underlined part: $-P \log \frac{1}{Q}$

KL Projection

- What does it mean?
  - When we maximize the log-likelihood of the model, we are minimizing the KL divergence between the empirical distribution and the model P(w|d) .

PLSA via EM

- E-step: estimate posterior probabilities of latent variables, ("concepts")

$$\mathrm{P}(z \mid d, w) = \frac{P(d \mid z) P(w \mid z) P(z)}{\sum_{z'} P(d \mid z') P(w \mid z') P(z')}$$

Probability that the occurence of term w in document d can be "explained" by concept z

- M-step: parameter estimation based on expected statistics.

$$\mathrm{P}(w \mid z) \propto \underbrace{\sum_{d} n(d, w) P(z \mid d, w)}$$

how often is term *w* associated with concept *z*

$$\mathrm{P}(d \mid z) \propto \underbrace{\sum_{w} n(d, w) P(z \mid d, w)}$$

how often is document *d* associated with concept *z*

$$\mathrm{P}(z) \propto \underbrace{\sum_{d, w} n(d, w) P(z \mid d, w)}$$

probability of concept *z*

Tempered EM

- The aspect model tend to over-fit easily.
  - Think about the number of free parameters we need to learn.
  - Entropic regularization based Tempered EM
  - E-Step is modified as follows:

$$\mathrm{P}(z \mid d, w) = \frac{[P(d \mid z) P(w \mid z) P(z)]^{\beta}}{\sum_{z'} [P(d \mid z') P(w \mid z') P(z')]^{\beta}}$$

  - Part of training data are held-out for internal validation. Best β is chosen based on this validation process.

Fold-in Queries/New Documents

- Concepts are not changed from the original training data.
- Only p(z|d) is changed, p(w|z) are the same in M-step.
- However, when we fix the concepts for new documents we are not getting the generative model any more.

PLSA Summary

- Optimal decomposition relies on likelihood function of multinomial sampling, which corresponds to a minimization of KL divergence between the empirical distribution and the model.
- Problem of polysemy is better addressed.
- Directions in the PLSA are multinomial word distributions.
- EM approach gives local solution.
- Possible to do the model selection and complexity control.
- Number of parameters increases linearly with number of documents.
- Not a generative model for new documents.

Link Analysis Techniques

- Motivations
  - The number of pages that could reasonably be returned as relevant is far too large for a human
  - identify those relevant pages that are the most authoritative
  - Page content is insufficient to define authoritativeness
  - Exploit hyperlink structure to assess and quantify authoritativeness

Hypertext Induced Topic Search (HITS)

- Associate two numerical scores with each document in a hyperlinked collection: authority score and hub score
  - **Authorities:** most definitive information sources (on a specific topic)
  - **Hubs**: most useful compilation of links to authoritative documents
- A good hub is a page that points to many good authorities; a good authority is a page that is pointed to by many good hubs

Iterative Score Computation

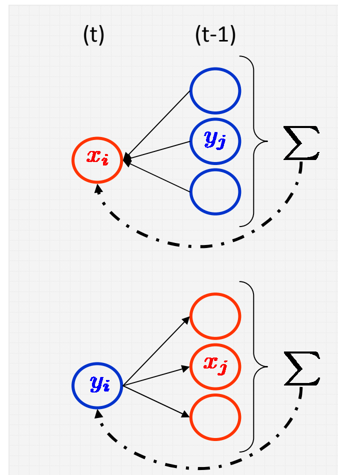
- Translate mutual relationship into iterative update equations

Authority scores

$$x_i^{(t)} \propto \sum_{j:(j,i)\in E} y_j^{(t-1)}$$

Hub scores

$$y_i^{(t)} \propto \sum_{j:(i,j)\in E} x_j^{(t-1)}$$



Matrix Notation

- Adjacency Matrix A

$$\mathbf{A} = (a_{ij}), \quad a_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

- Scores can be computed as follows:

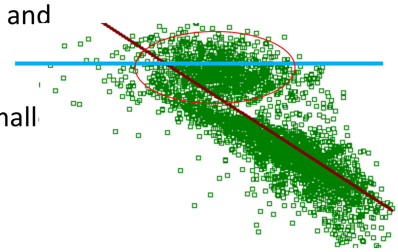
$$\mathbf{x}^{(t)} \propto \mathbf{A}^T \mathbf{y}^{(t-1)}, \qquad \mathbf{y}^{(t)} \propto \mathbf{A}\,\mathbf{x}^{(t-1)}$$

HITS Summary

- Compute query dependent authority and hub scores.
- Computational tractable (due to base set subgraph).
- Sensitive to Web spam (artificially increasing hub and authority weight, consider a highly interconnected set of sites).
- Dominant topic in base set may not be the intended one.
- Converge to the largest principle component of the adjacency matrix.

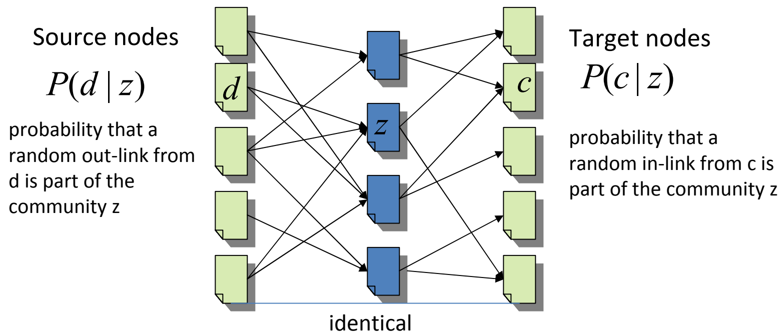
PHITS

- Probabilistic version of HITS.
- We try to find out the web communities from the Co-citation matrix.
- Loading on eigenvector in the case of HITS does not necessarily reflect the authority of document in community.
- HITS uses only the largest eigenvector and this is not necessary the principal community.
- What about smaller communities? (small eigenvectors) They can be still very important.
- Mathematically equivalent as PLSA

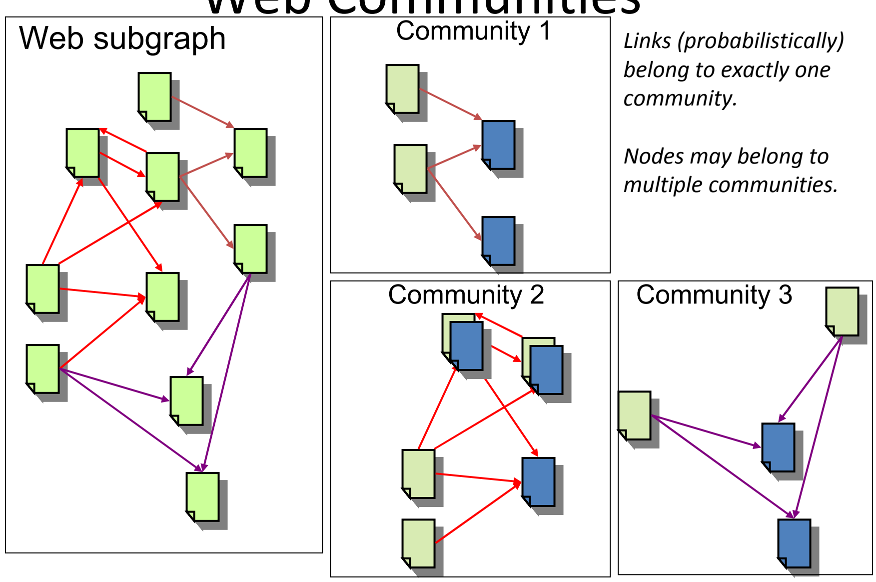


Finding Latent Web Communities

- Web Community: densely connected bipartite subgraph
- Probabilistic model pHITS: $P(d,c) = \sum_z P(z)P(d \mid z)P(c \mid z)$



Web Communities



Links (probabilistically) belong to exactly one community.

Nodes may belong to multiple communities.

PHITS: Model

- $\xrightarrow{P(d)} d \xrightarrow{P(z|d)} z \xrightarrow{P(c|z)} c$
- Add latent "communities" between documents and citations
- Describe citation likelihood as:

$$P(d,c) = P(d)P(c|d), \quad \text{where}$$
$$P(c|d) = \sum_z P(c|z)P(z|d)$$

- Total likelihood of citations matrix M:

$$L(M) = \prod_{(d,c) \in M} P(d,c)$$

- Process of building a model is transformed into a likelihood maximization problem.

PHITS via EM

- E-step: estimate the expectation of latent "community".

$$P(z \mid d, c) = \frac{[P(d \mid z)P(c \mid z)P(z)]^{\beta}}{\sum_{z'} [P(d \mid z')P(c \mid z')P(z')]^{\beta}}$$

*Probability that the particular document –citation pair is "explained" by community **z***

- M-step: parameter estimation based on expected statistics.

$$P(c \mid z) \propto \underbrace{\sum_d n(d,c)P(z \mid d,c)}$$

how often is citation c associated with community ***z***

$$P(d \mid z) \propto \underbrace{\sum_w n(d,c)P(z \mid d,c)}$$

how often is document ***d*** associated with community ***z***

$$P(z) \propto \underbrace{\sum_{d,w} n(d,c)P(z \mid d,c)}$$

probability of community ***z***

Interpreting the PHITS Results

- Simple analog to authority score is P(c|z).
  - How likely a document c is to be cited from within the community z.
- P(d|z) serves the same function as hub score.
  - The probability that document *d* contains a citation to a given community *z*.
- Document classification using P(z|c).
  - Classify the documents according its community membership.
- Find characteristic document of a community with P(z|c) * P(c|z).

PHITS Issues

- Local optimal solution from EM.
  - Possible to use PCA solution as the seed.
- Manually set the number of communities.
  - Split the factor and use model selection criterion like AIC and BIC to justify the split.
  - Iteratively extract factors and stop when the magnitude of them is over the threshhold.

Problems with Link-only Approach (e.g. PHITS)

- Not all links are created by human.
- The top ranked authority pages may be irrelevant to the query if they are just well connected.
- Web Spam.

PLSA and PHITS

- Joint probabilistic model of document content (PLSA) and connectivity (PHITS).
- Able to answer questions on both structure and content.
- Likelihood is

$$\mathcal{L} = \sum_j \left[\alpha \sum_i \frac{N_{ij}}{\sum_{i'} N_{i'j}} \log \sum_k P(t_i|z_k)P(z_k|d_j) + (1-\alpha) \sum_l \frac{A_{lj}}{\sum_{l'} A_{l'j}} \log \sum_k P(c_l|z_k)P(z_k|d_j) \right]$$

- EM approach to estimate the probabilities.

Reference Flow

- Two factor spaces $\vec{z}_m$ $\vec{z}_n$
- Documents d_i d_j
- Reference Flow between $\vec{z}_m$ $\vec{z}_n$

  $$f_{mn} = \sum_{i,j:A_{ij} \neq 0} P(d_i|\vec{z}_m)P(d_j|\vec{z}_n)$$

- This can be useful to create a better web crawler.
  - First locate the factor space of a new document using its content.
  - Use reference flow to compute the probability that this document could contain links to the factor space we are interested in.