Testing Dimension Reduction Methods for Text Retrieval

Pavel Moravec

Department of Computer Science, FEECS,
Technical University of Ostrava, Ostrava, Czech Republic

Databases, **Te**xts, **S**pecifications, and **O**bjects 2005

Motivation

- It is almost impossible to index text collections represented by vector model (VM) due to the curse of dimensionality.
- Text Collections represented in LSI model provide slightly better results in Text Retrieval than vector model, but are still hard to search efficiently and the computation of singular value decomposition is expensive.
- Other methods of dimension reduction are known, but were usually used on other types of collections and distance measures.

$\Longrightarrow$ We are looking for a method of dimension reduction, which will result in a small intrinsic dimensionality while preserving the precision and recall of original vector space model as much as possible.

Outline

Outline

1. Introduction
   - Vector Model
   - LSI Model
2. Other Dimension Reduction Methods
   - Random Projection
   - Approximate LSI Calculation
   - FastMap
3. Projection Properties
   - Intrinsic Dimensionality
   - Projection Stress
4. Qualitative Measures
5. Experimental Results
   - Experiment Setup
   - Analytical Results
   - Evaluation Results

Vector Model

Document D_i

- modeled with a vector d_i in vector space
- j-th coordinate of d_i represents a weight of j-th term in D_i

Document Collection (Corpus)

- represented as a term-by-document matrix A
- A is very sparse (max 1% nonzero values)
- high dimensionality of document vectors (= No. of terms)

Term-by-document Matrix Example

term \ doc.	D_1	D_2	D_3	...	D_m
database	0	0.48	0.05		0.70
vector	0.23	0	0.23		0
...				⋱	
image	0	0	0.10		0.54

Queries in Vector Model

Similarity Function

- classifies similarity of a document vector d_i to a query vector q
  - cosine measure $\mathrm{SIM}_{\cos}(d_i, d_j)$ – cosine of vector deviation
- query evaluation made by using the similarity function
  - range queries (similarity threshold r_q)
  - k-NN queries (searching for k nearest neighbors)

Curse of Dimensionality

Most indexing structures degenerate in higher dimensions – it is cheaper to use sequential scan. This phenomenon is called "curse of dimensionality" and we try to lessen its impact by dimension reduction techniques.

Queries in Vector Model

Similarity Function

- classifies similarity of a document vector d_i to a query vector q
  - cosine measure $\mathrm{SIM}_{\cos}(d_i, d_j)$ – cosine of vector deviation
- query evaluation made by using the similarity function
  - range queries (similarity threshold r_q)
  - k-NN queries (searching for k nearest neighbors)

Curse of Dimensionality

Most indexing structures degenerate in higher dimensions – it is cheaper to use sequential scan. This phenomenon is called “curse of dimensionality” and we try to lessen its impact by dimension reduction techniques.

LSI Model – Singular Value Decomposition

Singular Value Decomposition

The term-by-document matrix A is decomposed to

$$A = U\Sigma V^T$$

- U, V^T – column-orthonormal matrices of singular vectors
- Σ is a diagonal matrix of singular values

k-reduced Singular Value Decomposition (SVD)

- First dimensions are the most important ones $\implies$ we use only k first dimensions; the rest is discarded as “semantic noise”
- Thus we construct the k-reduced SVD as

$$A = U\Sigma V^T \approx A_k = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix}$$

LSI Model – Singular Value Decomposition

Singular Value Decomposition

The term-by-document matrix A is decomposed to

$$A = U \Sigma V^T$$

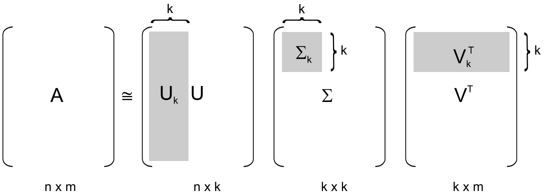
- U, V^T – column-orthonormal matrices of singular vectors
- Σ is a diagonal matrix of singular values

k-reduced Singular Value Decomposition (SVD)

- First dimensions are the most important ones $\implies$ we use only k first dimensions; the rest is discarded as "semantic noise"
- Thus we construct the k-reduced SVD as

$$A = U \Sigma V^T \approx A_k = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix}$$

LSI Model – Latent Semantic Indexing



Latent Semantics Indexing Provides

- latent semantics – hidden connections between both terms and documents determined on documents' content
- dimensionality reduction (e.g. from hundreds of thousands to several hundreds)
- the search is term independent, it is concept-based
  - LSI model partially solves the problem of synonymy and homonymy

LSI Model – Model Description

Document and Query Representation

- Concept-by-document matrix
  $D_k = \Sigma_k V_k^T \times D'_k = V_k^T$
- Reduced query vector
  $q_k = U_k^T q \times q'_k = \Sigma_k^{-1} U_k^T q$
- Projection matrix
  $P_k = U_k^T \times P'_k = \Sigma_k^{-1} U_k^T$

Term Similarity

- Concept-by-term matrix
  $T_k = U_k \Sigma_k \times T'_k = U_k$

LSI Model – Model Description

Document and Query Representation

- Concept-by-document matrix
  $D_k = \Sigma_k V_k^T \;\times\; D'_k = V_k^T$
- Reduced query vector
  $q_k = U_k^T q \;\times\; q'_k = \Sigma_k^{-1} U_k^T q$
- Projection matrix
  $P_k = U_k^T \;\times\; P'_k = \Sigma_k^{-1} U_k^T$

Term Similarity

- Concept-by-term matrix
  $T_k = U_k \Sigma_k \;\times\; T'_k = U_k$

Outline

Random Projection

Random Projection Description

- Projection into subspace of suitable dimension by randomly generated projection matrix R.
- Elements of projection matrix are independent random variables with a zero mean unit variance distribution.
- Scaling of projected vector by $\sqrt{\frac{d}{k}}$ is applied after projection to preserve Euclidean distances (for cosine measure scaling is not needed).
- Random projection may not be a projection if matrix R is not orthogonal.
- Orthogonality and normality don't have to be enforced
  - projection vectors lengths are already around 1 with γ-distribution of their squares.
  - random directions might be close to orthogonal, $R^T R$ approximates identity matrix.

Random Projection

"Classic" Random Projection

Projection matrix contains Gaussian-distributed random values

$$r_{ij} \in N(0,1)$$

Simple Random Projection

Projection matrix contains integer values $r_{ij} \in \{-1, 0, 1\}$, computation of projection can be done by addition, final normalization is applied. Two variants exist:

$$r_{ij} = \sqrt{3}. \begin{cases} -1 & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability } \frac{2}{3} \\ +1 & \text{with probability } \frac{1}{6}. \end{cases} \qquad r_{ij} = \begin{cases} -1 & \text{with probability } \frac{1}{2} \\ +1 & \text{with probability } \frac{1}{2}. \end{cases}$$

Approximate LSI Calculation

Approximate LSI by Random Projection

Rank-2k LSI can be used after random projection into a high-enough dimension l, $l > 2k$ and recovers almost as much as classical rank-k LSI. The upper bound is

$$||A - B_{2k}||_F^2 \leq ||A - A_k||_F^2 + 2\varepsilon ||A||_F^2$$

where

$$||J||_F^2 = \sum_{\substack{i=1\ldots n \\ k=1\ldots m}} J_{ik}^2$$

is Frobenius norm.

Approximate LSI by Monte-Carlo Method

Calculates rank-k SVD on randomly-chosen $s \times s$ submatrices.

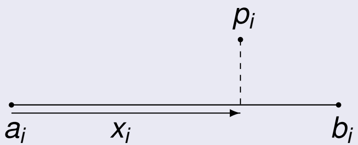
FastMap

FastMap Description

- A pivot-based technique of dimension reduction, suitable for Euclidean spaces.
- Pivots in original & reduced space are recorded, projection uses the second step of projection algorithm only.

Main Ideas

- All points are projected onto a line connecting two (probably) most distant objects.
- Cosine law is used to calculate distance in current dimension and the distance function is modified.



FastMap

Projection Algorithm

1. A random point c_0 is chosen.,
2. point b_i having maximal distance $\delta(c_i, b_i)$ from c_i is chosen, and based on it we select the point a_i with maximal distance $\delta(b_i, a_i)$,
3. we iteratively repeat step 2 with $c_{i+1} = a_i$ (authors: $5\times$).

In second step, we use the cosine law to calculate position of each point on line joining a and b. The coordinate x_i of point p_i is calculated as

$$x_i = \frac{\delta^2(a_i,p_i)+\delta^2(a_i,b_i)-\delta^2(b_i,p_i)}{2\delta(a_i,b_i)}$$

and the distance function for next reduction step is modified to

$$\delta'^2(p'_i, p'_j) = \delta^2(p_i, p_j) - (x_i - x_j)^2$$

FastMap

Projection Algorithm

1. A random point c_0 is chosen.,
2. point b_i having maximal distance $\delta(c_i, b_i)$ from c_i is chosen, and based on it we select the point a_i with maximal distance $\delta(b_i, a_i)$,
3. we iteratively repeat step 2 with $c_{i+1} = a_i$ (authors: $5\times$).

In second step, we use the cosine law to calculate position of each point on line joining a and b. The coordinate x_i of point p_i is calculated as

$$x_i = \frac{\delta^2(a_i,p_i)+\delta^2(a_i,b_i)-\delta^2(b_i,p_i)}{2\delta(a_i,b_i)}$$

and the distance function for next reduction step is modified to

$$\delta'^2(p_i', p_j') = \delta^2(p_i, p_j) - (x_i - x_j)^2$$

Outline

1. Introduction
   - Vector Model
   - LSI Model
2. Other Dimension Reduction Methods
   - Random Projection
   - Approximate LSI Calculation
   - FastMap
3. Projection Properties
   - Intrinsic Dimensionality
   - Projection Stress
4. Qualitative Measures
5. Experimental Results
   - Experiment Setup
   - Analytical Results
   - Evaluation Results

Intrinsic Dimensionality

Consequences of the "Curse of Dimensionality"

- Exact search methods are inefficient for high dimensions
- In tree-like structures it is usually reflected by huge region overlaps
  - each region overlaps the query region $\Longrightarrow$ the search deteriorates to sequential search

Intrinsic Dimensionality of Metric Spaces

- Generalization of the "curse of dimensionality"
- Definition, based on distance distribution histograms:

$$\rho(\mathbb{S}, d) = \frac{\mu^2}{2\sigma^2}$$

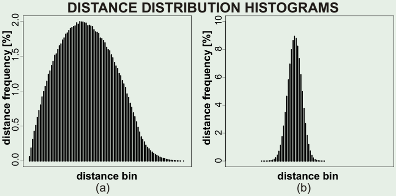
where μ and σ^2 are the mean and the variance of the dataset distance distribution (according to a metric d).

Intrinsic Dimensionality

Interpretation

- higher intrinsic dimensionality $\Longrightarrow$ less structured collection $\Longrightarrow$ harder filtering of irrelevant objects
- the goal is to decrease the intrinsic dimensionality in order to obtain a better performance of searching methods

Distance Distribution Histograms Example



DDHs indicating (a) low (b) high intrinsic dimensionality

Projection Stress

Projection Stress Definition

- How well are all distances preserved after projection.
- The stress of projection f is calculated as

$$stress = \sqrt{\frac{\left(\sum_{i,j=1}^{m} (d'(f(x_i), f(x_j)) - d(x_i, x_j)\right)^2}{\sum_{i,j=1}^{m} d^2(x_i, x_j)}}$$

where d is the distance function in original and d' in projected space.

Consequences

1. The lower the stress, the better; $stress = 0 \implies$ the projection did not change the distances.
2. When the distances are scaled by some coefficient, the projection stress may not give us meaningful results.

Outline

Qualitative Measures

Used Qualitative Measures

(P) Precision – fraction of retrieved relevant documents in retrieved ones.

(R) Recall – fraction of retrieved relevant documents in all relevant ones.

Usage of Rank Lists

1. Rank list – ordered query result (from the most similar document).
2. To calculate the precision and recall independently on required thresholds, the interpolated precision on 11 standard recall levels (0.0, 0.1, . . . , 1.0) is calculated. The results are presented by P-R curves.
3. We calculated the mean average precision and compared it with classic vector model.

Outline

Experiment Setup

Text Collection

- We used a subset of TREC collection, part of Los Angeles Times (LATimes) articles (years 1989 and 1990), with 16,889 documents and 49,689 terms after filtration.
- Projection of matrix A into dimensions between 50 and 1000 (depending on given method) was calculated.
- Random Projection calculation was fastest ($\approx 100\times$ than FastMap), LSI calculation the slowest ($\approx 5\times$ than FM)

Queries

- 50 TREC-8 ad-hoc queries were used for qualitative evaluation.
- Query projection: Random Projection was the fastest ($\approx 2\times$ than LSI), FastMap the slowest ($\approx 2.5\times$ LSI)

Analytical Results

Intrinsic Dimensionality (31.8 for VM)

k	LSI	FastMap	RP	RP+LSI
50	25.1	0.2	53.3	46.8
100	51.1	0.5	100.2	93.9
250	121.1	0.9	217.1	206.4
500	–	–	343.7	329.7
1000	–	–	489.3	–

Projection Stress

k	LSI	FastMap	RP	RP+LSI
50	0.210	0.978	0.296	0.247
100	0.224	0.978	0.284	0.259
250	0.242	0.980	0.282	0.270
500	–	–	0.279	0.275
1000	–	–	0.278	–

Evaluation Results – Average Precision

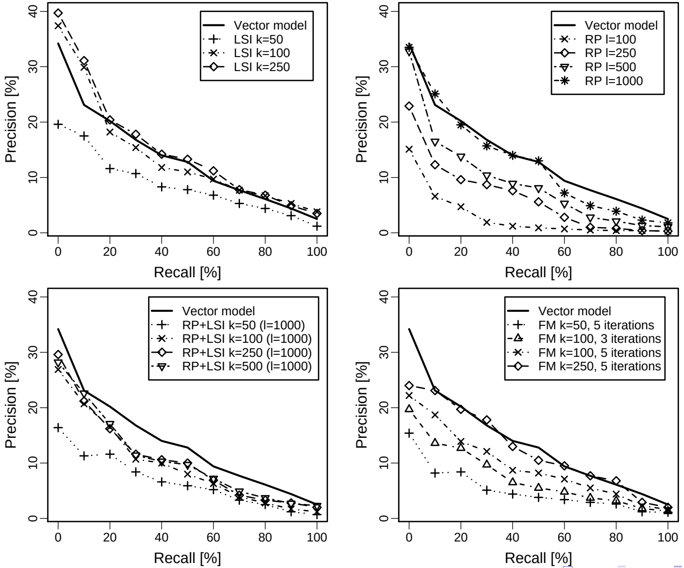
Mean Average Precision

k	LSI	FastMap	RP	RP+LSI
50	128%	31%	13%	85%
100	155%	58%	24%	98%
250	112%	80%	37%	79%
500	–	–	59%	77%
1000	–	–	74%	–

Average Precision at 100% Recall

k	LSI	FastMap	RP	RP+LSI
50	117%	74%	80%	113%
100	124%	88%	88%	118%
250	102%	90%	89%	105%
500	–	–	98%	101%
1000	–	–	100%	–

Evaluation Results – P-R curves



Conclusion

Outcome

- We compared several dimension reduction methods from different angles.
- While LSI is the best one, we run into problems when trying to decompose a greater collection. The FastMap (or approximate LSI) may suffice.

Future Work

- We may speed-up the FastMap calculation by sampling or some heuristics.
- In future, we plan to use greater collection (omitting LSI) and newly proposed methods such as SparseMap and MetricMap.

References

M. Berry and M. Browne.
Understanding Search Engines, Mathematical Modeling and Text Retrieval. Siam, 1999.

C. Böhm, S. Berchtold, and D. Keim.
Searching in High-Dimensional Spaces – Index Structures for Improving the Performance of Multimedia Databases.

E. Chávez, G. Navarro, R. Baeza-Yates and J. Marroquín.
Searching in metric spaces.

C. Papadimitriou, H. Tamaki, P. Raghavan and S. Vempala.
Latent semantic indexing: A probabilistic analysis.

E. M. Voorhees and D. Harman.
Overview of the sixth text REtrieval conference (TREC-6).