Latent Semantic Analysis for large document sets.

Laurence A. F. Park Kotagiri Ramamohanarao

ARC Centre for Perceptive and Intelligent Machines in Complex Environments
Department of Computer Science and Software Engineering
The University of Melbourne

Overview

- Latent semantic analysis (LSA) is a simple concept with firm mathematical foundations, but it has many problems.
- We propose a query expansion which utilises LSA and overcomes most of the problems.

Outline

Outline

Document index

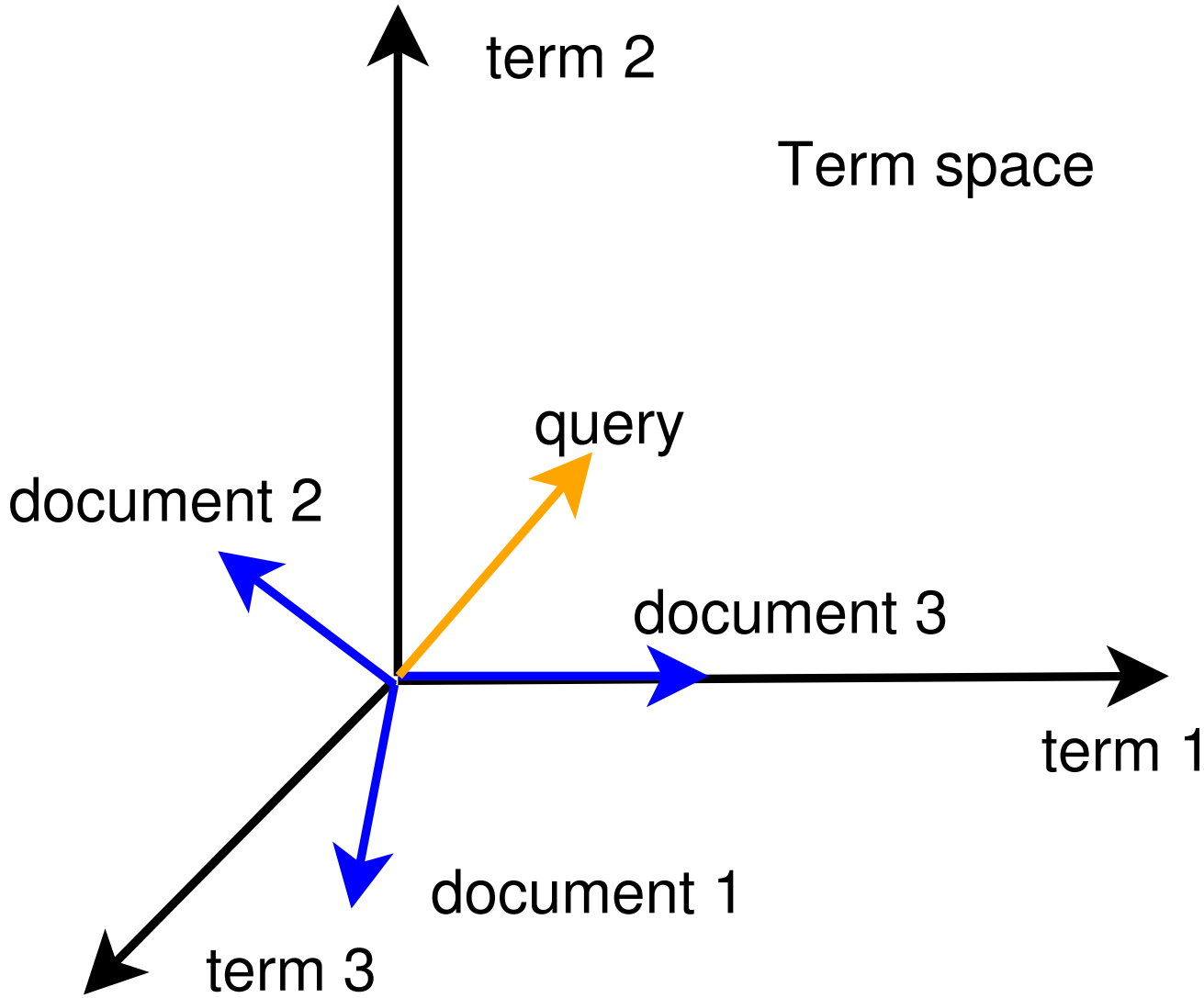
Document index containing document vectors:

	term 1	term 2	term 3
document 1	1	0	2
document 2	0	1	1
document 3	1	0	0

Query vector:

	term 1	term 2	term 3
query	1	1	0

Document and query vectors

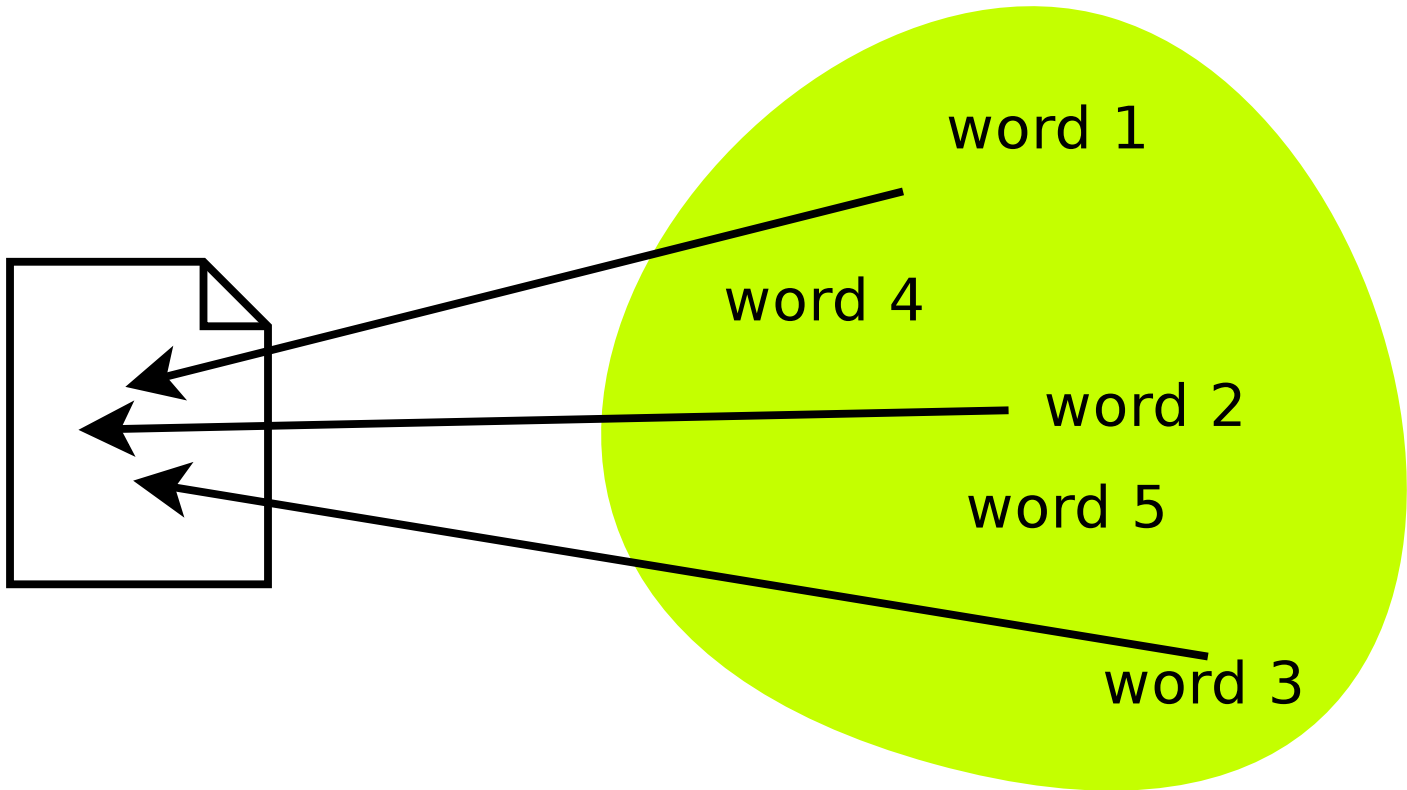


The term space

- The vector space and probabilistic model treats terms as independent entities.
- To find documents on a specific topic, we must provide the correct key terms.

VSM Document creation process

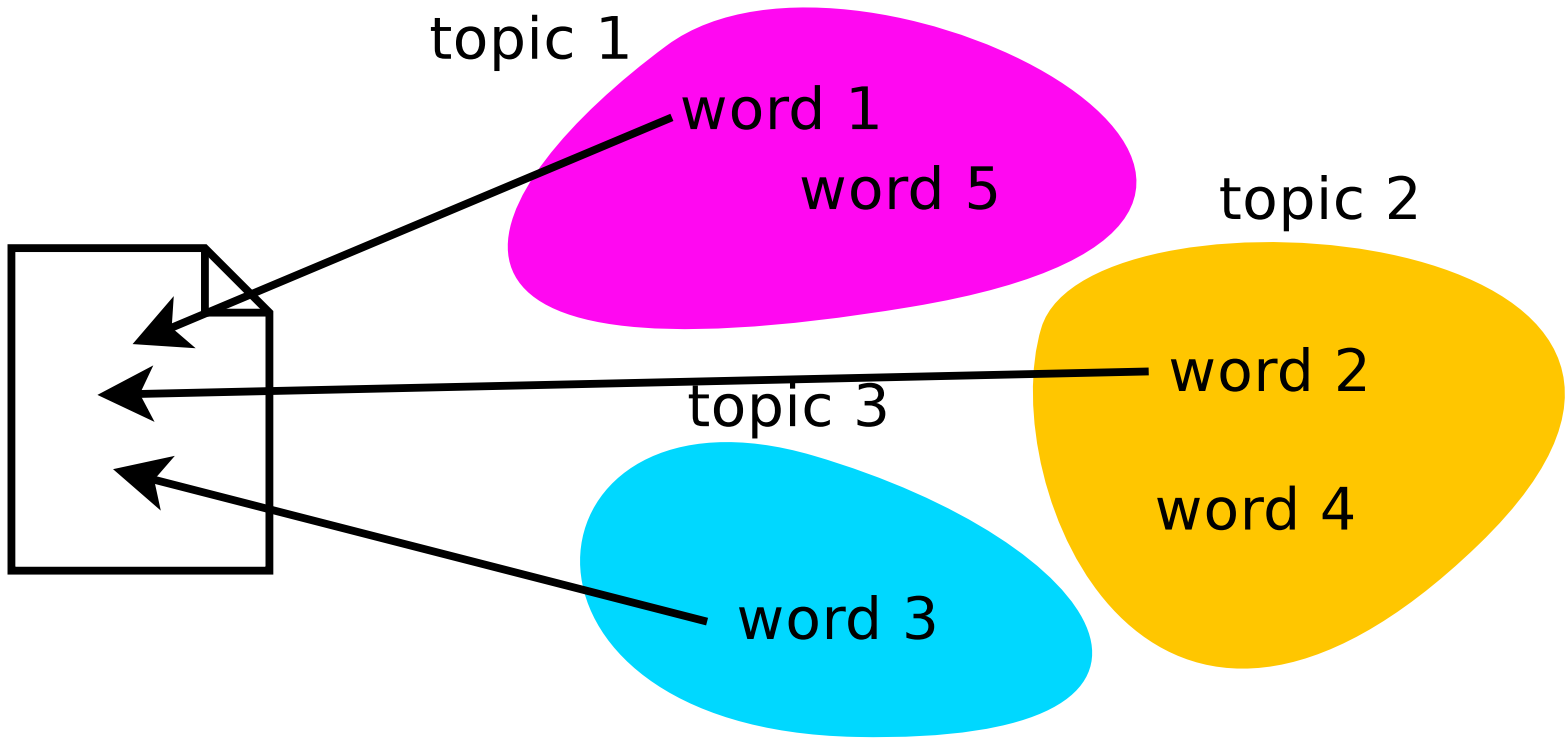
1. Author has an idea.
2. Idea is put to paper in the form of words.



This implies that if any word was changed, the document would not convey the same idea.

LSA Document creation process

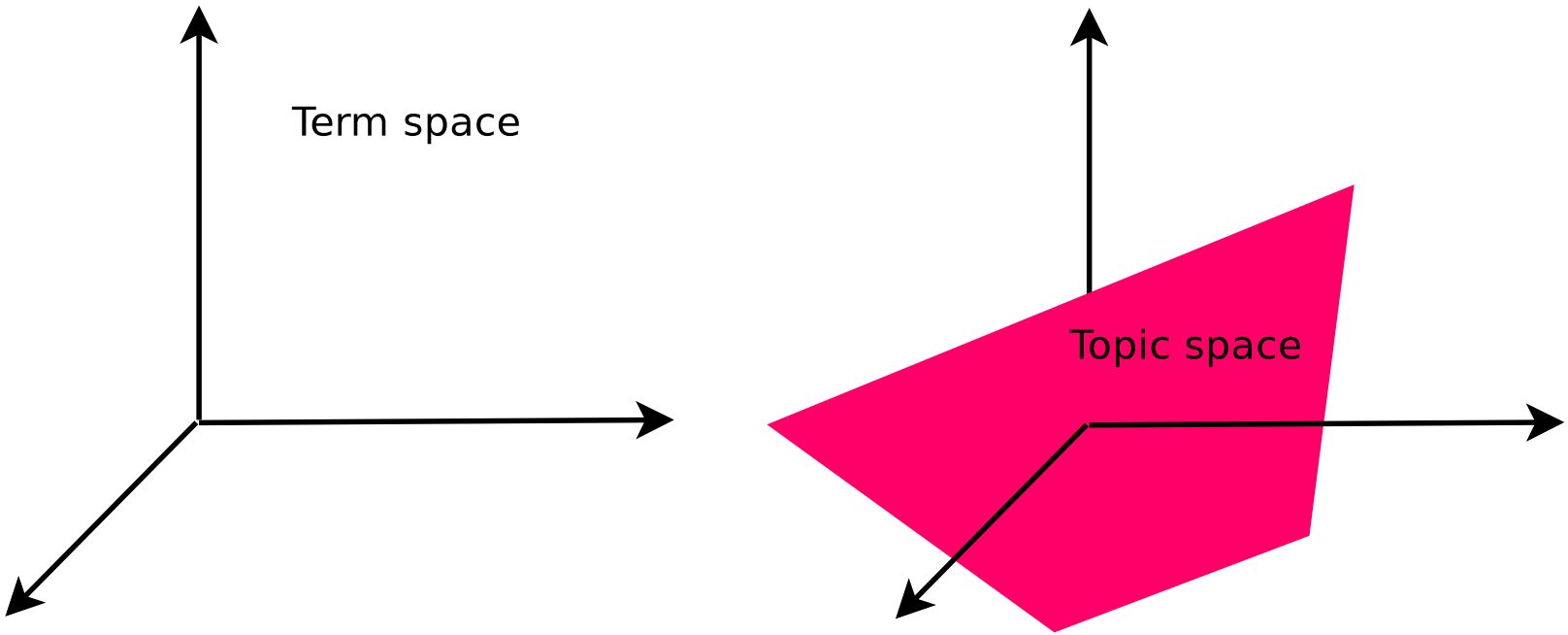
1. Author has an idea.
2. Idea to divided into topics.
3. Topic is put to paper in the form of words taken from the topic pool.



If words are changed, the document conveys the same idea as long as the new word was taken from the same topic pool.

Creation of space

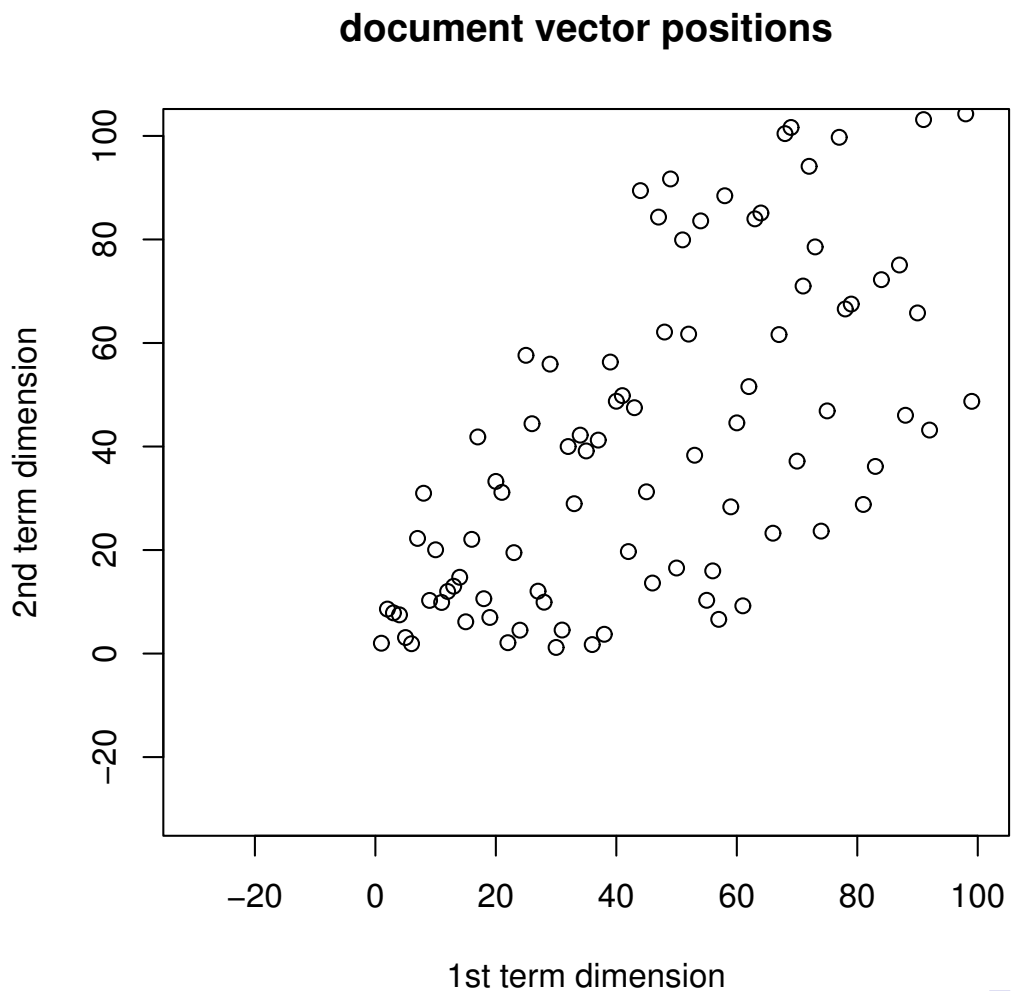
1. A topic is a set of similar words
2. Topics can be found by clustering
3. LSA uses reduction of dimension to cluster the words into terms



Outline

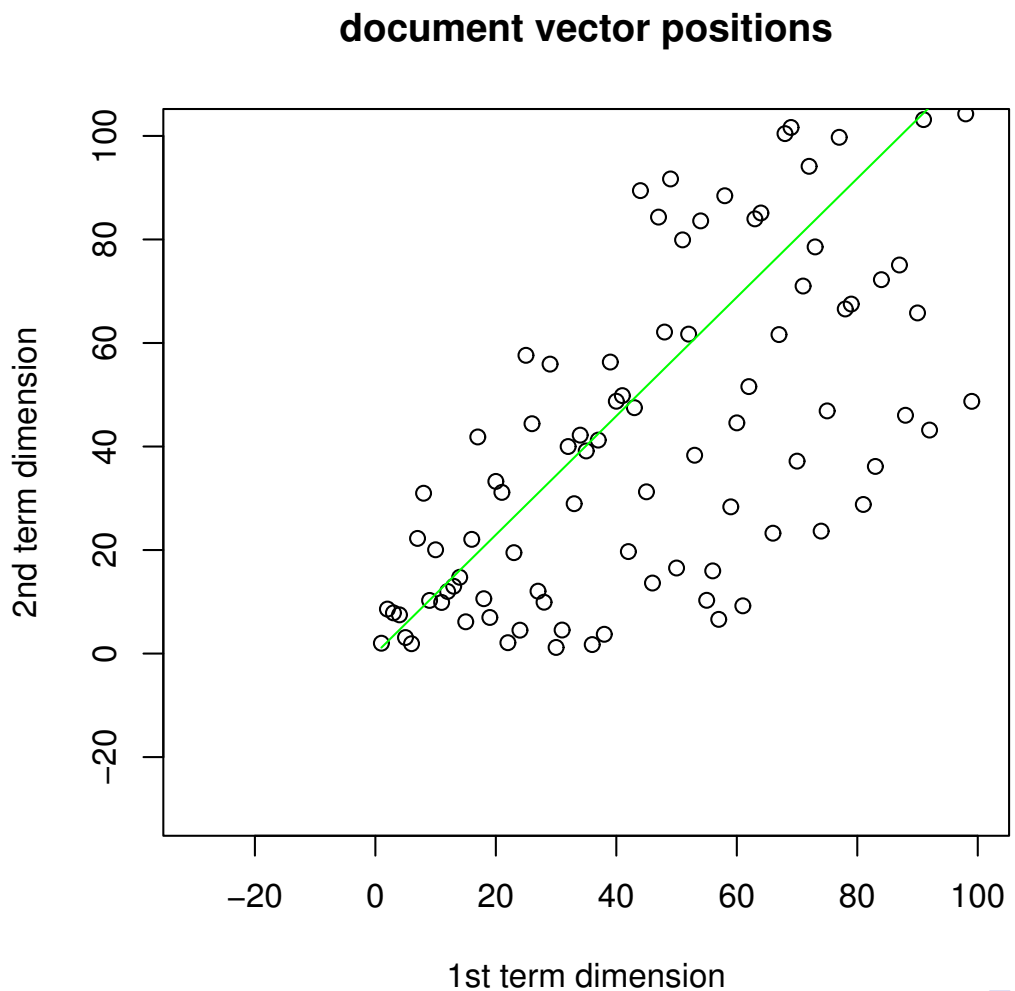
LSA concepts

- The dimensions of greatest variance holds the most information.
- The dimensions of small variance are considered noise.



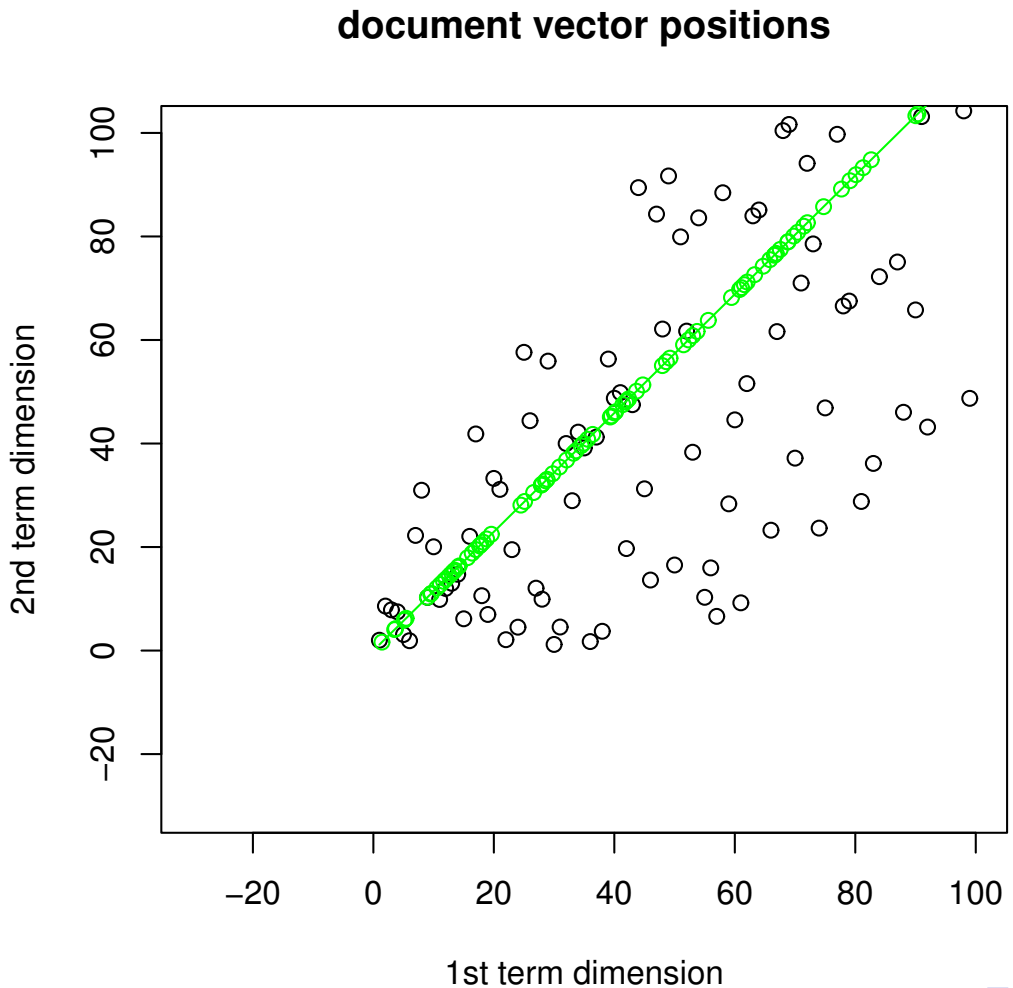
LSA concepts

- The dimensions of greatest variance holds the most information.
- The dimensions of small variance are considered noise.



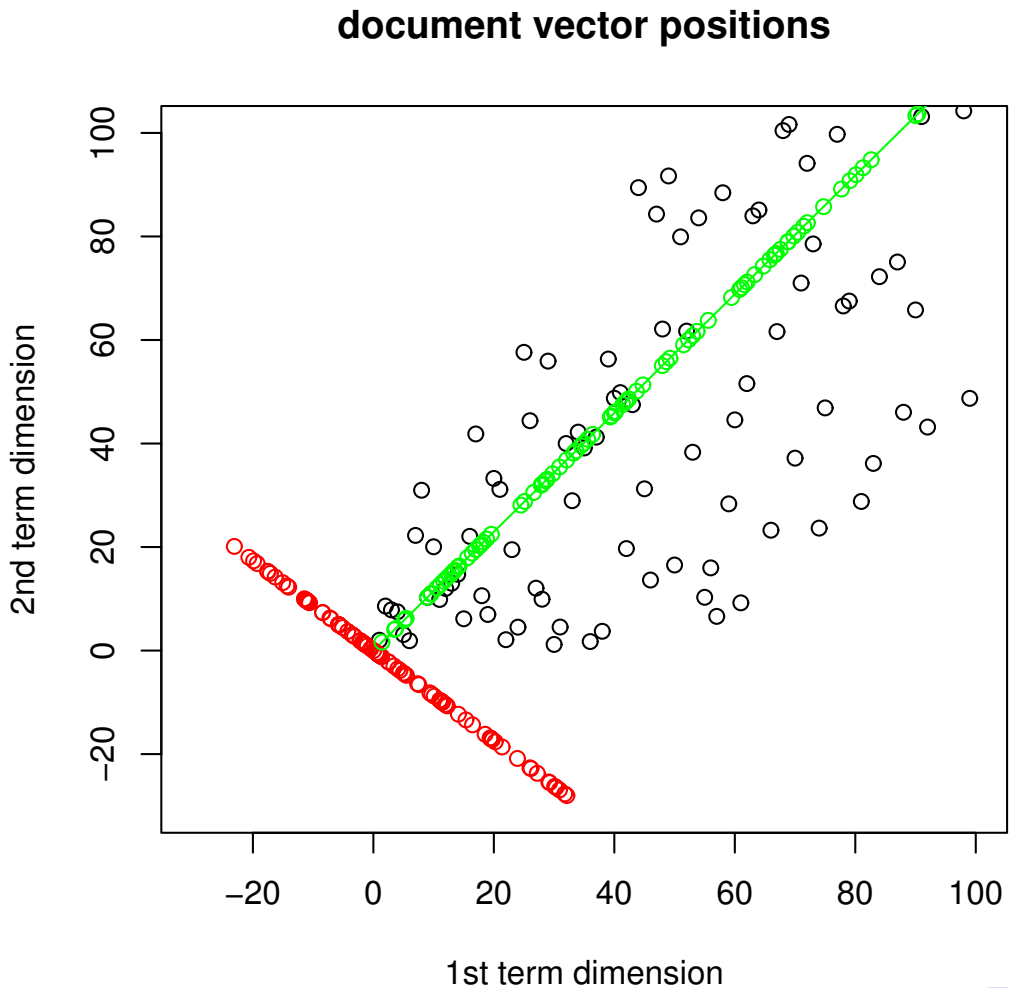
LSA concepts

- The dimensions of greatest variance holds the most information.
- The dimensions of small variance are considered noise.



LSA concepts

- The dimensions of greatest variance holds the most information.
- The dimensions of small variance are considered noise.



LSA process

LSA index building:

1. Construct a space based on the largest s orthogonal norms of the document index.
2. Find the best approximation of every document vector in the new topic space.

LSA process

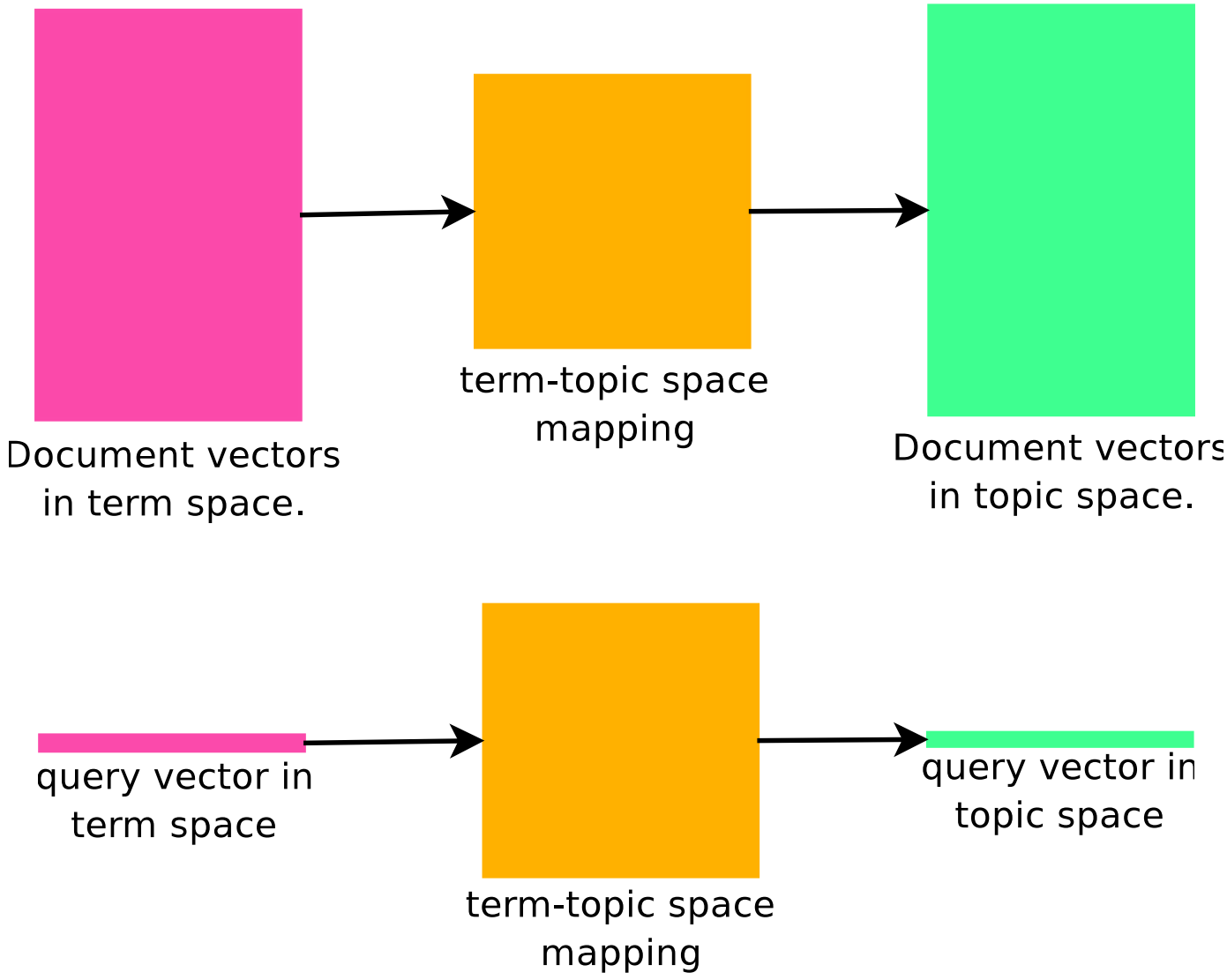
LSA index building:

1. Construct a space based on the largest s orthogonal norms of the document index.
2. Find the best approximation of every document vector in the new topic space.

LSA query:

1. Calculate the best approximation of the query vector in the topic space.
2. Calculate the query-document similarity score by using the inner product of the vectors in the topic space.

LSA concepts

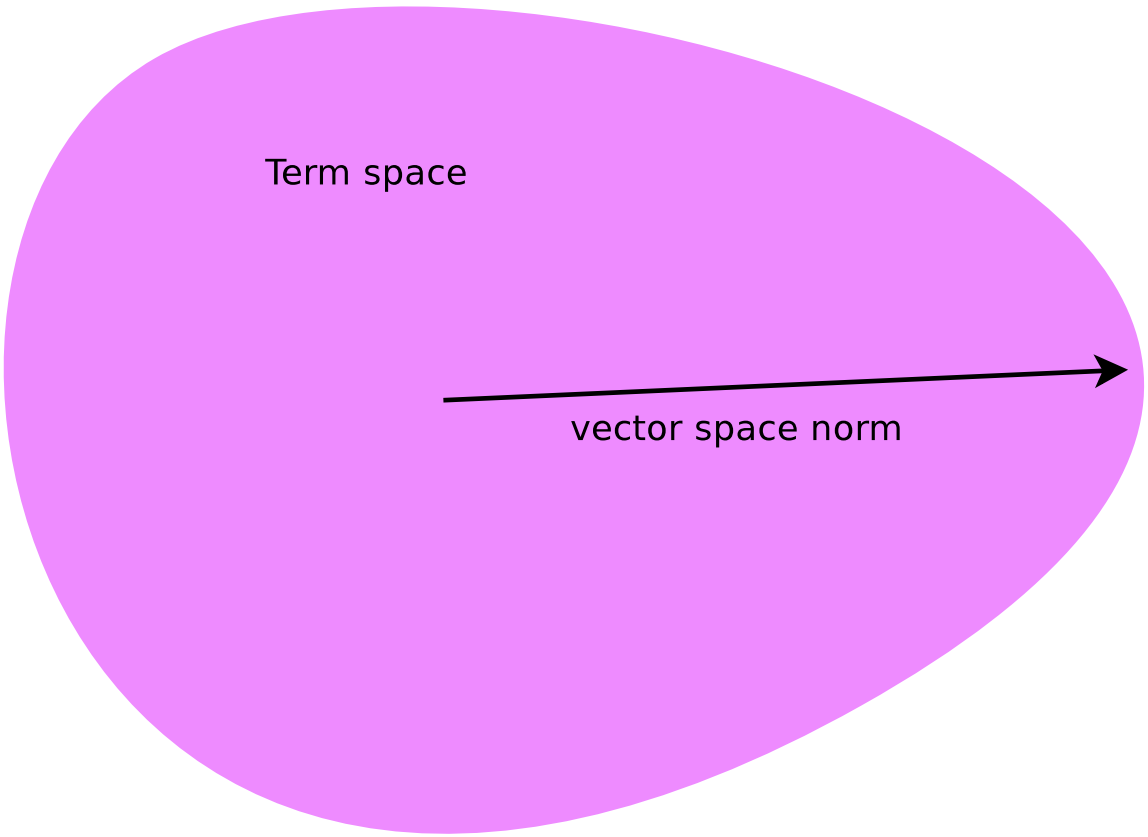


Dimension of greatest variance

Document matrix norm

$$\sigma = \|A\|_2 \equiv \max_{\mathbf{x}} \left\|A\mathbf{x}'\right\|_2 \qquad \|\mathbf{x}\|_2 = 1 \tag{1}$$

$$\mathbf{v} = \operatorname*{argmax}_{\mathbf{x}} \left\|A\mathbf{x}'\right\|_2 \tag{2}$$



Calculating the dimension

The vector $\mathbf{v}$ provides the most information about the document set of any unit vector existing in the document space.

Calculating $\mathbf{v}$

$$\sigma = \max_{\mathbf{v}} \left\| A\mathbf{v}' \right\|_2 \tag{3}$$

$$\left\| A\mathbf{v}' \right\|_2^2 = \sigma^2 = \left(A\mathbf{v}'\right)' \left(A\mathbf{v}'\right) \tag{4}$$

$$= \mathbf{v}A'A\mathbf{v}' \tag{5}$$

$$\mathbf{v}\mathbf{v}' = 1 \tag{6}$$

$$\mathbf{v}\mathbf{v}'\sigma^2 = \mathbf{v}A'A\mathbf{v}' \tag{7}$$

$$\mathbf{v}'\sigma^2 = A'A\mathbf{v}' \tag{8}$$

Therefore σ^2 and $\mathbf{v}$ are an eigenvalue and eigenvector of $A'A$.

Singular Values

The eigenvectors and eigenvalues of $A'A$ are the singular vectors and singular values of A.

Calculating $\mathbf{v}$

$$
\begin{aligned}
A &= U'\Sigma V && (9)\\
A'A &= V'\Sigma UU'\Sigma V && (10)\\
&= V'\Sigma^2 V && (11)\\
A'AV' &= V'\Sigma^2 && (12)
\end{aligned}
$$

or:

$$A'A\mathbf{v}' = \mathbf{v}'\sigma^2$$

Therefore the norm of a matrix is equal to its largest singular value.

Singular Value Decomposition Properties

Singular value decomposition

$$A = U'\Sigma V \tag{13}$$

- U and V contain the left and right orthonormal singular vectors.
- $U'U = V'V = I$
- $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, 0, \ldots, 0)$

Second largest dimension

To find the second largest dimension:

- remove the largest dimension
- find the largest dimension in the new space

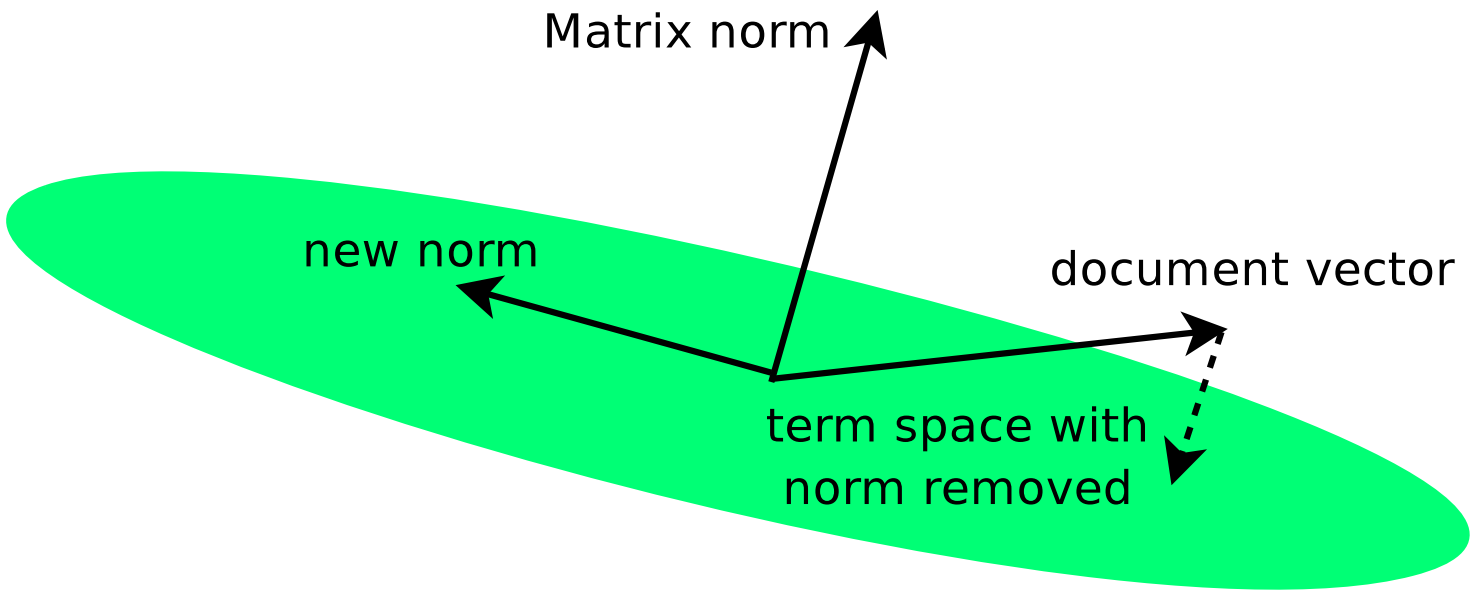
To remove the largest dimension:

- project the document vectors onto this dimension
- subtract it from the original document set

$$A_1 = A - A\mathbf{v}'\mathbf{v} \tag{14}$$

Selection of matrix norm

- After removing the dimension corresponding to the matrix norm, the next largest norm becomes the largest remaining dimension norm.
- This is the next largest singular value.



Choosing singular values

- To choose the r dimensions that best approximate the original matrix, we select the top r singular vector and corresponding singular values.

Outline

LSA Example

Example

Simple document set

- The stone is large enough
- Large stones are fast
- Fast stones are not smooth enough

$$\mathbf{d} = [\ \textit{stone}\ \textit{large}\ \textit{enough}\ \textit{fast}\ \textit{smooth}\ ] \qquad A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

σ_i	$\mathbf{v}_i$				
2.715	−0.6357	−0.3991	−0.4362	−0.4362	−0.2367
1.276	0.1118	0.7123	−0.2444	−0.2444	−0.6005
1	0	0	0.7071	−0.7071	0
0	0.6621	−0.5667	−0.09537	−0.09537	−0.4714
0	0.3807	0.1101	−0.4908	−0.4908	0.6009

LSA Example

Example

Simple document set

- The stone is large enough
- Large stones are fast
- Fast stones are not smooth enough

$$\mathbf{d} = [\ \textit{stone}\ \textit{large}\ \textit{enough}\ \textit{fast}\ \textit{smooth}\ ] \qquad A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

σ_i	$\mathbf{v}_i$				
2.715	−0.6357	−0.3991	−0.4362	−0.4362	−0.2367
1.276	0.1118	0.7123	−0.2444	−0.2444	−0.6005
1	0	0	0.7071	−0.7071	0
0	0.6621	−0.5667	−0.09537	−0.09537	−0.4714
0	0.3807	0.1101	−0.4908	−0.4908	0.6009

LSA Example

We can see from the singular values that the matrix A has rank = 3 and two redundant dimensions.

- Each singular value is associated to a singular vector.
- We choose the top two singular vectors as our new basis.
- The document vectors are mapped into our new space.

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \hat{V}' = \begin{bmatrix} -0.6357 & 0.1118 \\ -0.3991 & 0.7123 \\ -0.4362 & -0.2444 \\ -0.4362 & -0.2444 \\ -0.2367 & -0.6005 \end{bmatrix}$$

$$\hat{A} = A\hat{V}' = \begin{bmatrix} -1.471 & 0.5797 \\ -1.471 & 0.5797 \\ -1.745 & -0.9775 \end{bmatrix}$$

LSA Example - Inserting the Query

Example

Query

- Query = "fast stone"

To compare queries to document, they must both exist in the same space.

$$\mathbf{q} = [\ 1\ 0\ 0\ 1\ 0\ ]$$

$$\hat{\mathbf{q}} = \mathbf{q}\hat{V}' = \begin{bmatrix} -1.072 & -0.1326 \end{bmatrix}$$

LSA Example - Calculate scores

Documents and queries are compared in the topic space using the inner product.

$$
\begin{aligned}
\mathbf{s} &= \hat{A}\hat{\mathbf{q}}' \\
&= AV'(qV')' = \begin{bmatrix} -1.471 & 0.5797 \\ -1.471 & 0.5797 \\ -1.745 & -0.9775 \end{bmatrix} \begin{bmatrix} -1.072 \\ -0.1326 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.5 \\ 2 \end{bmatrix}
\end{aligned}
$$

Outline

Past LSA Results

- LSA showed improvement over the vector space model for the Cranfield (cran, cisi, cacm, med) data sets.
- LSA was used in TREC-3; 10% of the documents were used in the topic space calculations. Displayed small improvements.
- Own experiments have shown:
  - an improvement over the base VSM using the small Cranfield document sets ($\sim$ 1000 documents).
  - very poor precision using larger sets as in AP2, WSJ2 ($\sim$ 100,000 documents)

Outline

LSA Precision, Storage and Speed

Problems with precision:

- LSA on large document set produces poor precision.
- Synonymy is covered, polysemy is not.

LSA Precision, Storage and Speed

Problems with precision:

- LSA on large document set produces poor precision.
- Synonymy is covered, polysemy is not.

Problems with query speed:

- Query in topic space has very few zero elements.
- Query process involves examining every term in the index.
- Document and query vectors in the topic space may have negative elements.
- Cannot use accumulator.

LSA Precision, Storage and Speed

Problems with precision:

- LSA on large document set produces poor precision.
- Synonymy is covered, polysemy is not.

Problems with query speed:

- Query in topic space has very few zero elements.
- Query process involves examining every term in the index.
- Document and query vectors in the topic space may have negative elements.
- Cannot use accumulator.

Problems with document vector storage:

- Document vectors in the topic space have very few zero elements.
- Document vector elements are real values.
- Inverted index is useless.

Leaving the topic space

These problems cause us not to be able to use LSA on large document sets. We must leave the topic space, but we don't have to leave its effects.
We have used:

- documents: term $\longrightarrow$ topic
- query: term $\longrightarrow$ topic

We have seen that the topic space is too dense

Leaving the topic space

To achieve the same results, we can change to:

- documents: term $\longrightarrow$ topic $\longrightarrow$ term
- query: term

provides sparse query (fast), but dense documents (large storage)
or:

- documents: term
- query: term $\longrightarrow$ topic $\longrightarrow$ term

provides sparse documents (compact), dense query (slow)

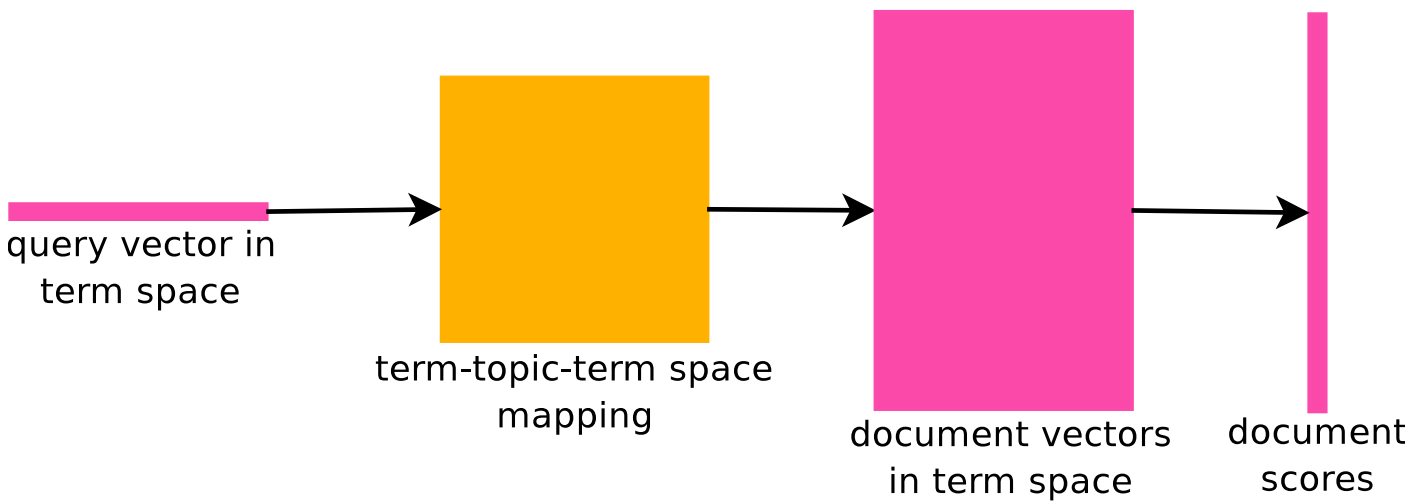
Outline

Query Expansion

The latter option is a LSA query expansion. The expansion is performed by:

1. Map the query to the reduced dimensional topic space.
2. Map the query back into the term space

$$\begin{aligned} \mathbf{s} &= (AV')(qV')' && (15) \\ &= A(V'V)q' && (16) \\ &= AMq' && (17) \end{aligned}$$



Query map example

Example

Simple document set

- The stone is large enough
- Large stones are fast
- Fast stones are not smooth enough

$$\mathbf{d} = [\ \textit{stone large enough fast smooth}\ ] \qquad A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

σ_i	$\mathbf{v}_i$				
2.715	−0.6357	−0.3991	−0.4362	−0.4362	−0.2367
1.276	0.1118	0.7123	−0.2444	−0.2444	−0.6005
1	0	0	0.7071	−0.7071	0
0	0.6621	−0.5667	−0.09537	−0.09537	−0.4714
0	0.3807	0.1101	−0.4908	−0.4908	0.6009

Query map example

Example

Simple document set

- The stone is large enough
- Large stones are fast
- Fast stones are not smooth enough

$$\mathbf{d} = [\ \textit{stone large enough fast smooth}\ ] \qquad A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

σ_i	$\mathbf{v}_i$				
2.715	−0.6357	−0.3991	−0.4362	−0.4362	−0.2367
1.276	0.1118	0.7123	−0.2444	−0.2444	−0.6005
1	0	0	0.7071	−0.7071	0
0	0.6621	−0.5667	−0.09537	−0.09537	−0.4714
0	0.3807	0.1101	−0.4908	−0.4908	0.6009

Query map example

We can see from the singular values that the matrix A has rank $=$ 3 and two redundant dimensions.

- Each singular value is associated to a singular vector.
- We choose the top two singular vectors as our new basis.
- The basis is multiplied by its inverse to create our query map.

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \hat{V}' = \begin{bmatrix} -0.6357 & 0.1118 \\ -0.3991 & 0.7123 \\ -0.4362 & -0.2444 \\ -0.4362 & -0.2444 \\ -0.2367 & -0.6005 \end{bmatrix}$$

$$M = \hat{V}'\hat{V} = \begin{bmatrix} 0.4167 & 0.3333 & 0.25 & 0.25 & 0.08333 \\ 0.3333 & 0.6667 & 0 & 0 & -0.3333 \\ 0.25 & 0 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0 & 0.25 & 0.25 & 0.25 \\ 0.08333 & -0.3333 & 0.25 & 0.25 & 0.4167 \end{bmatrix}$$

Query map example - expanding the query

Example

Query

- Query = "fast stone"

To compare queries to document, they must both exist in the same space.

$$\mathbf{q} = [\ 1\ 0\ 0\ 1\ 0\ ]$$

$$\hat{\mathbf{q}}' = M\mathbf{q}' = \begin{bmatrix} 0.6667 & 0.3333 & 0.5 & 0.5 & 0.3333 \end{bmatrix}'$$

Query map example - Calculate scores

Documents and queries are compared in the term space using the inner product.

$$\mathbf{s} = A\hat{\mathbf{q}}' = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.6667 \\ 0.3333 \\ 0.5 \\ 0.5 \\ 0.3333 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.5 \\ 2 \end{bmatrix}$$

This provides the same score as in LSA, but we can identify the term relationships.

Examining the query map

We can now observe the term relationships.

	stone	large	enough	fast	smooth
stone	0.4167	0.3333	0.25	0.25	0.08333
large	0.3333	0.6667	0	0	-0.3333
enough	0.25	0	0.25	0.25	0.25
fast	0.25	0	0.25	0.25	0.25
smooth	0.08333	-0.3333	0.25	0.25	0.4167

Outline

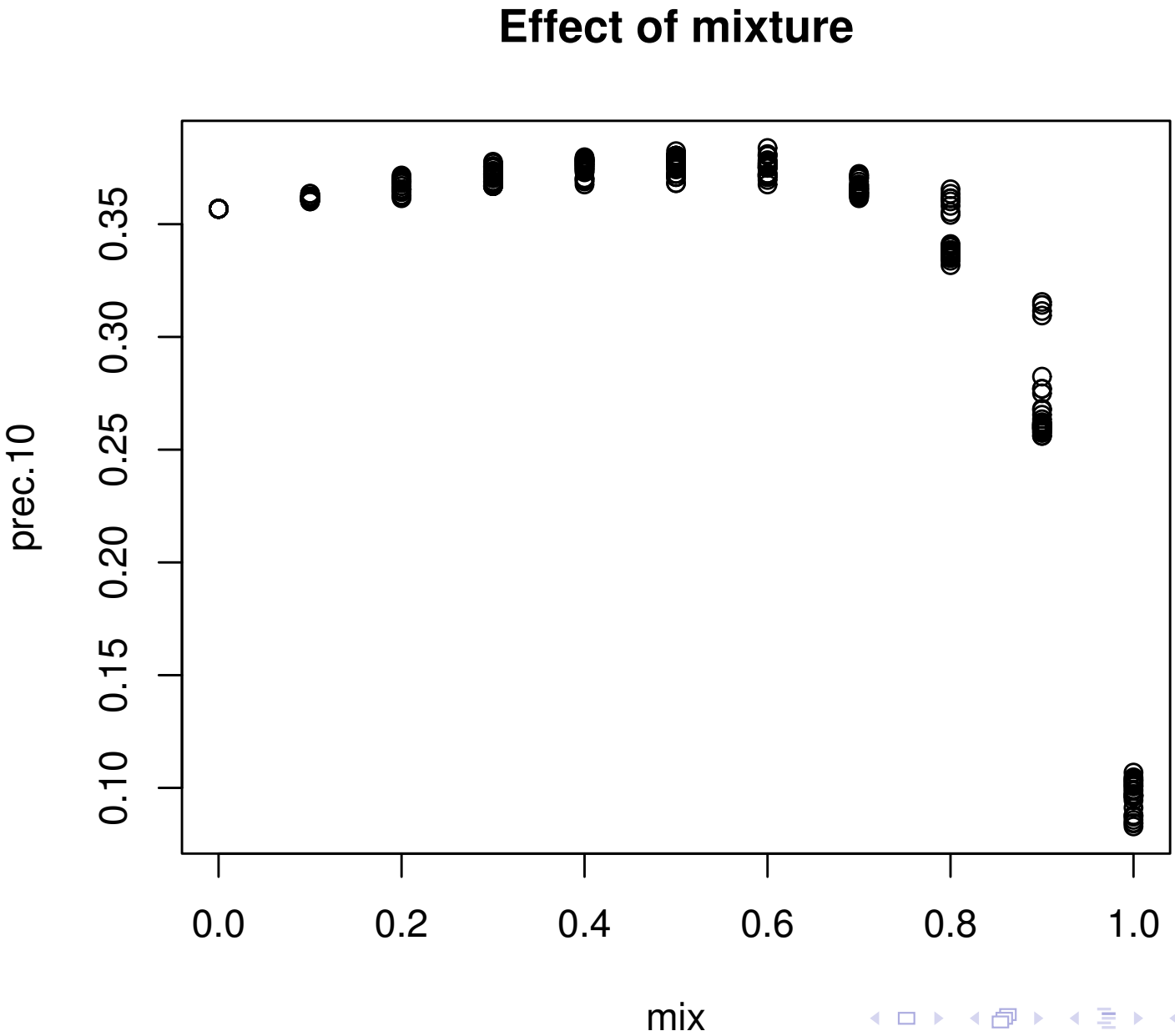
Mixing LSA with the VSM

- LSA was shown to under represent the query terms.
- One solution: mix the LSA and VSM scores.
- Can be combined into the one mapping.

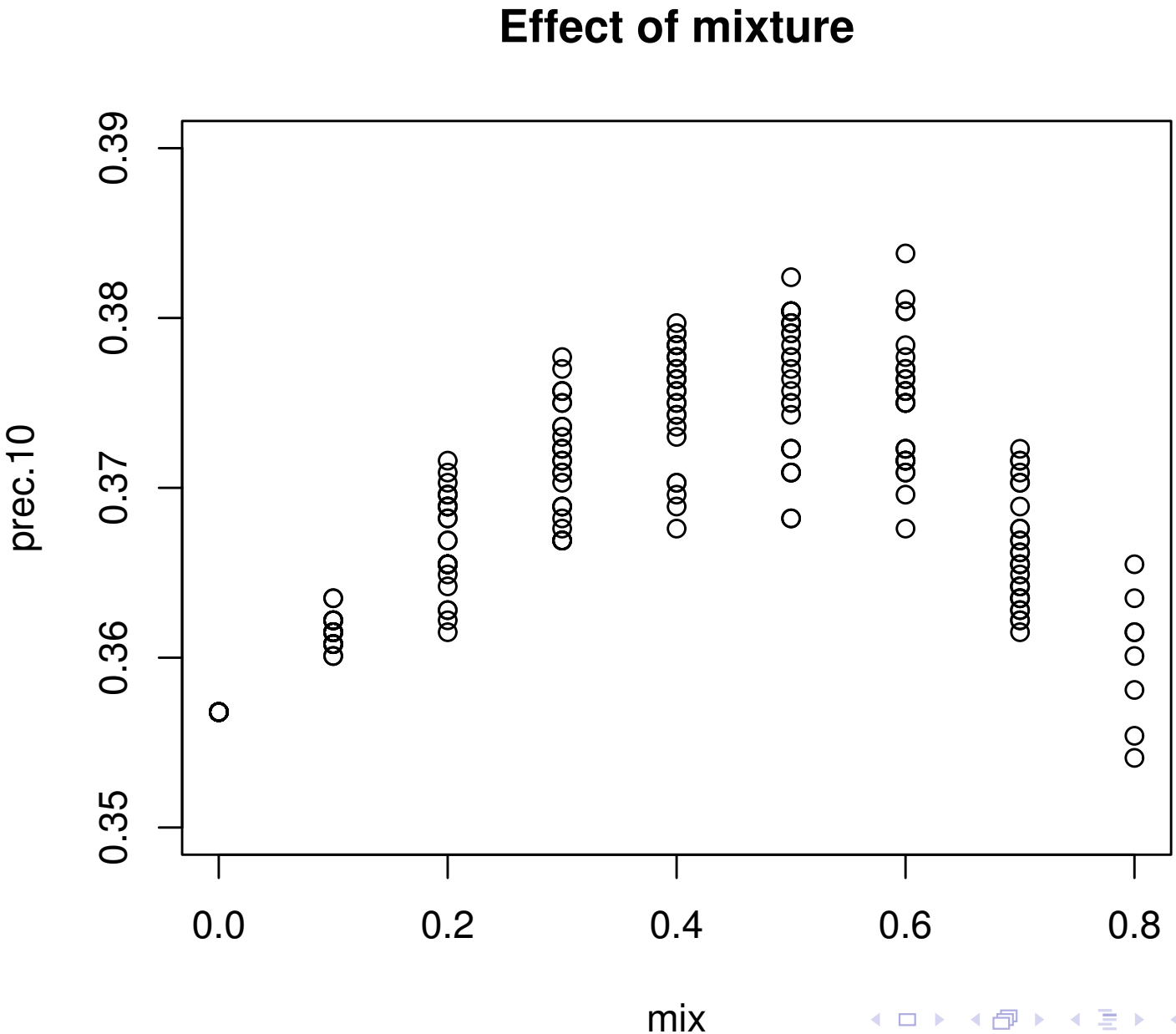
$$
\begin{aligned}
\mathbf{s} &= \alpha AMq' + (1-\alpha)Aq' && (18)\\
&= \alpha AMq' + (1-\alpha)AIq' && (19)\\
&= A(\alpha M + (1-\alpha)I)q' && (20)\\
&= AM_\alpha q' && (21)
\end{aligned}
$$

Note: $\alpha = 0 \Rightarrow$ s = VSM, $\alpha = 1 \Rightarrow$ s = LSA.

Results: LSA + VSM



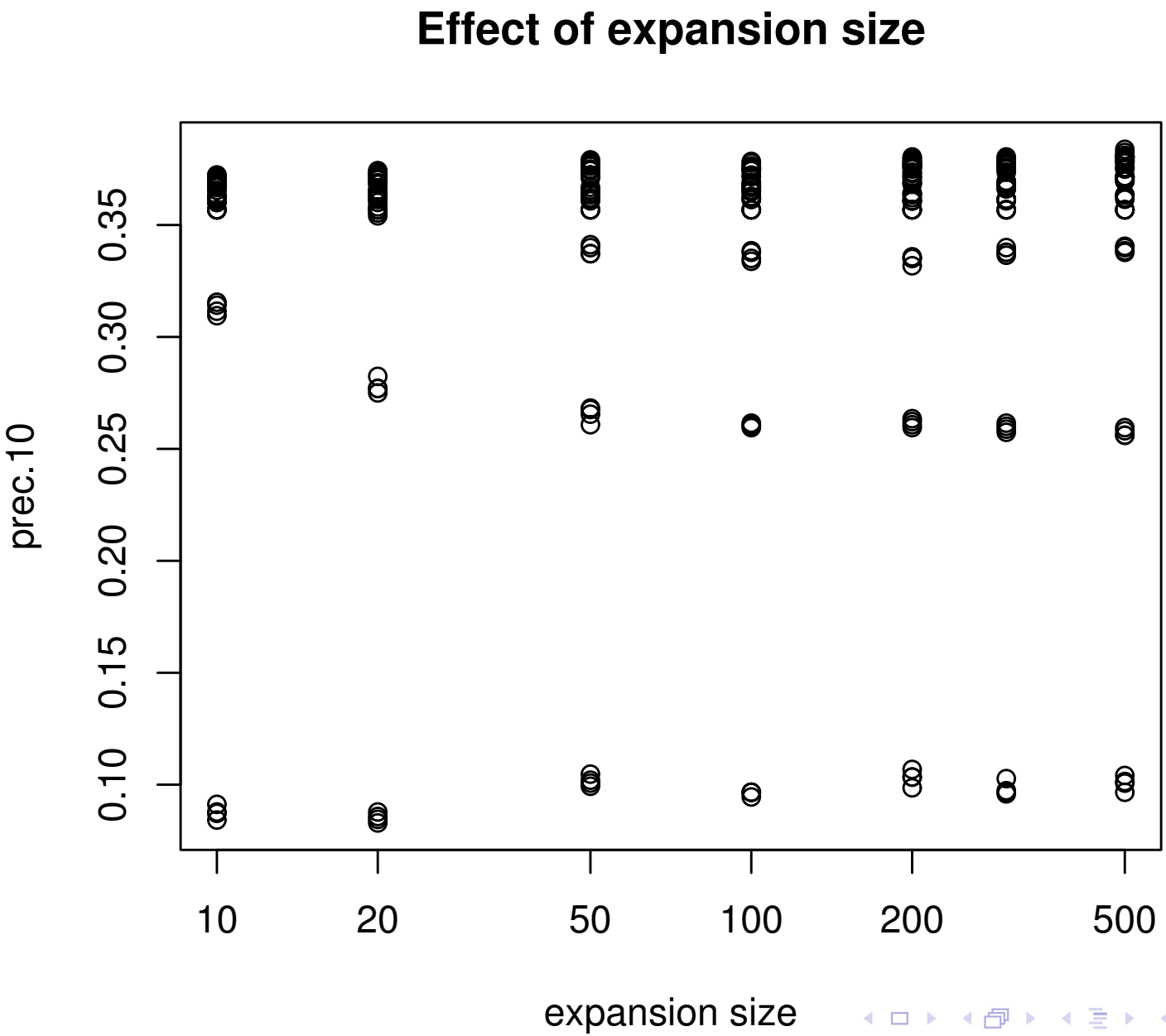
Results: LSA + VSM



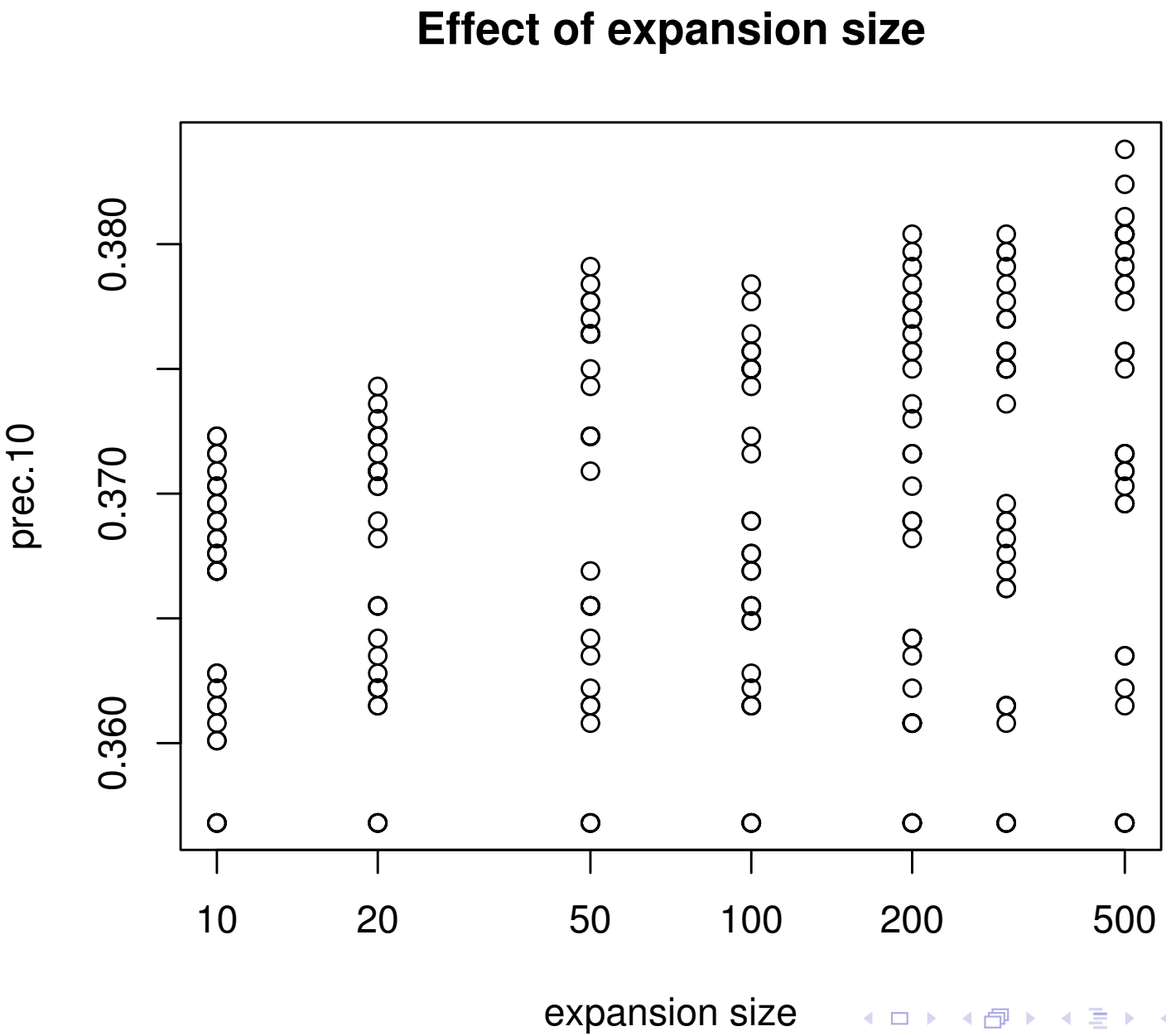
Expanded term selection

- Query times are long due to the dense query vector.
- Now that we have our query expanded into terms, we can select which terms we want and reduce the query time.
- This allows the mapping to be used by large document sets.

Results: Term selection



Results: Term selection



Expansion term set

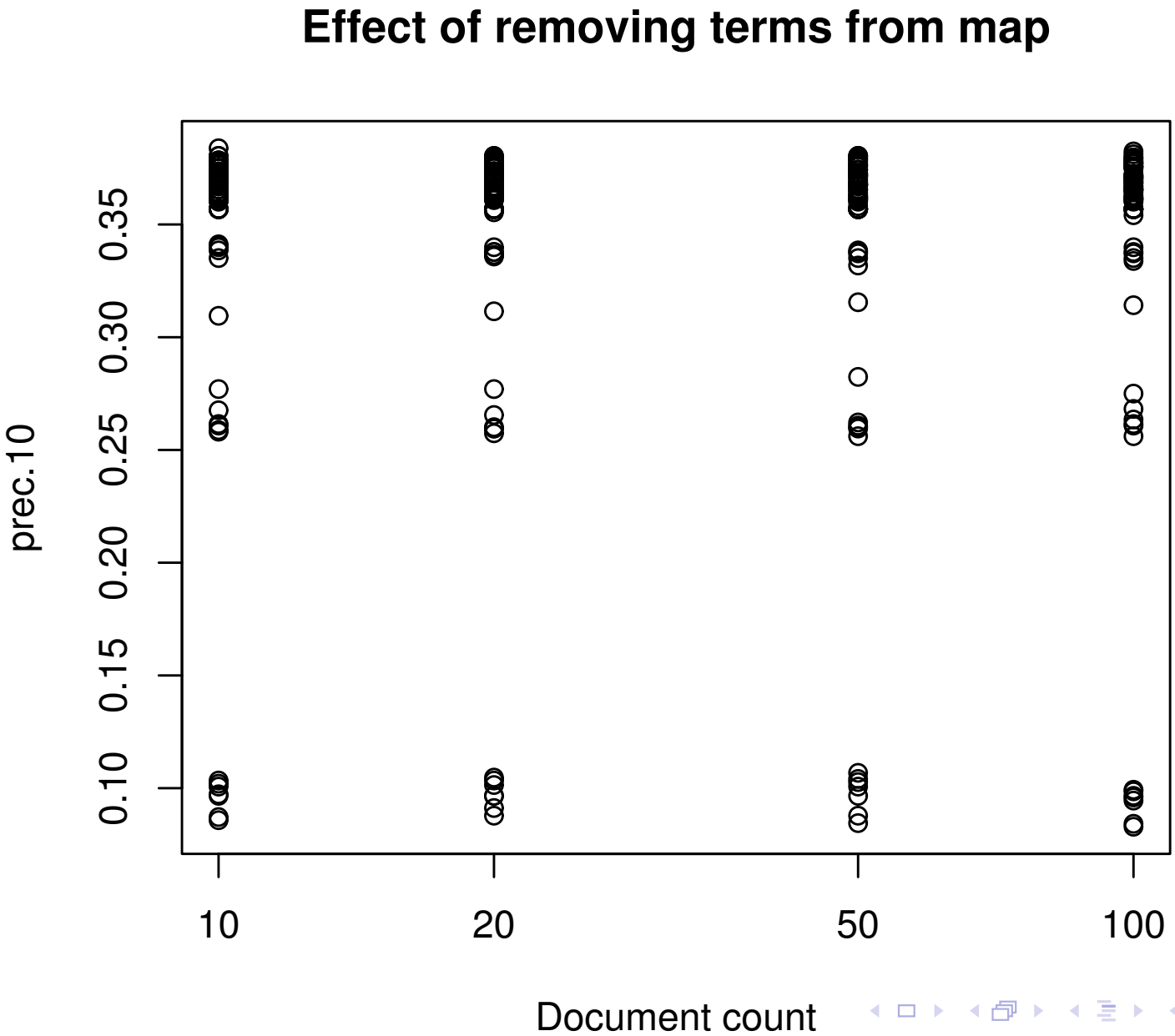
- If terms were removed from LSA, they will not contribute to the document score.
- By mixing LSA with the VSM we are guaranteed to have the query term in the expanded query.
- Therefore, we can be more selective about the LSA terms.
- This allows the mapping to be used by large document sets.

Hypothesis

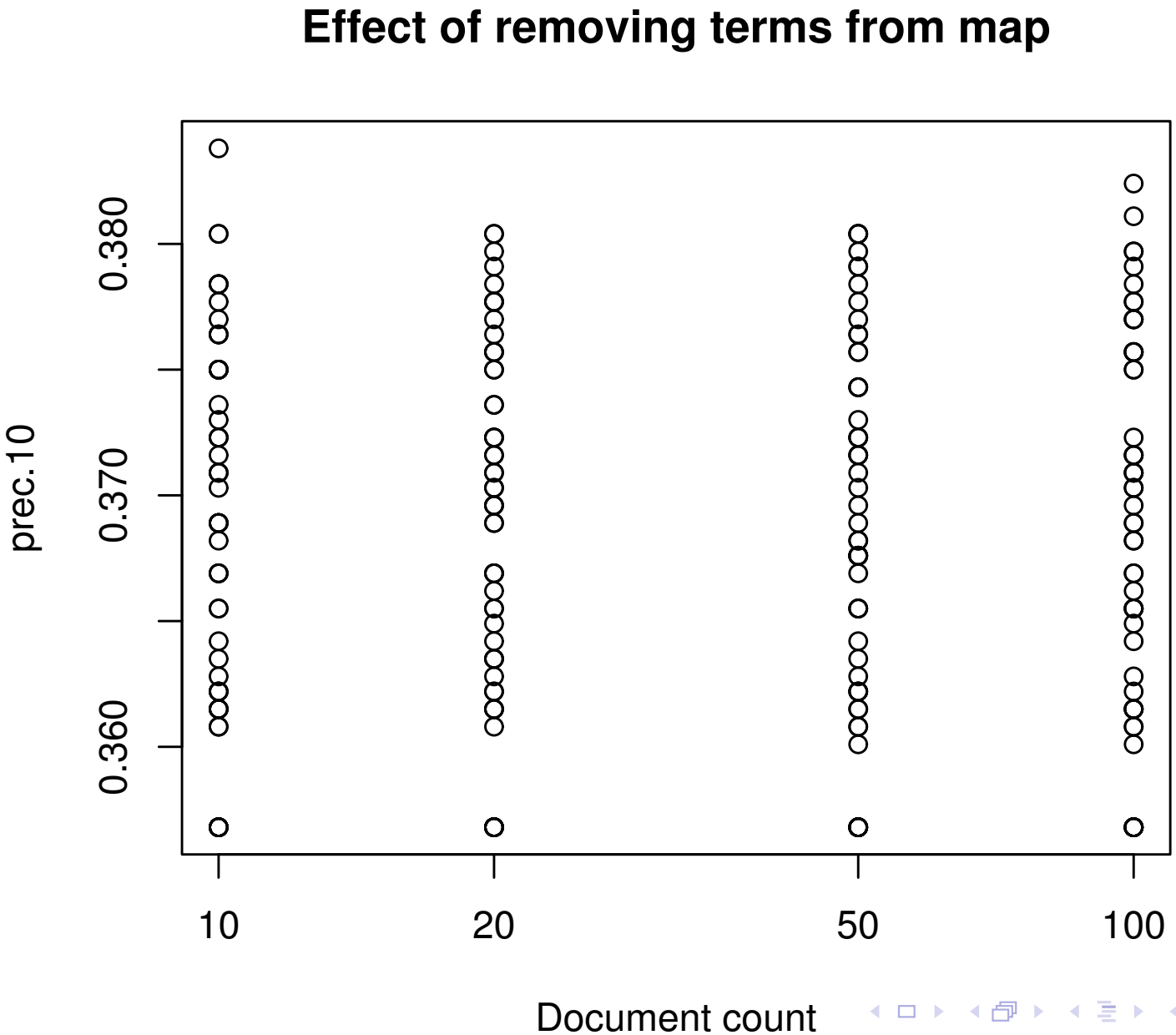
LSA is unable to construct effective term relationships with under sampled terms.

We experimented by removing those terms that appeared in less than N documents.

Results: Remove under sampled terms

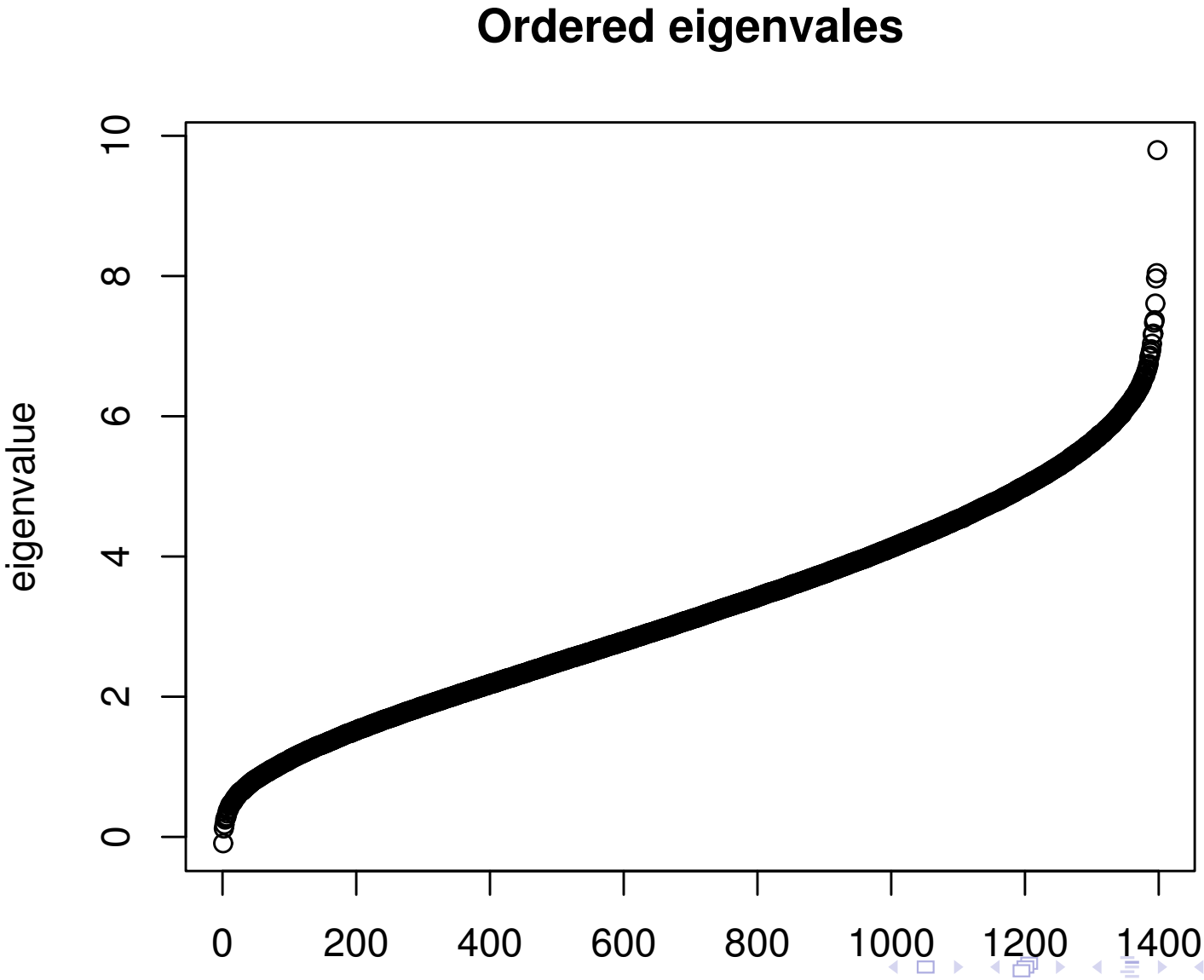


Results: Remove under sampled terms



Topic space size

Problem: how many eigenvalues do we choose?

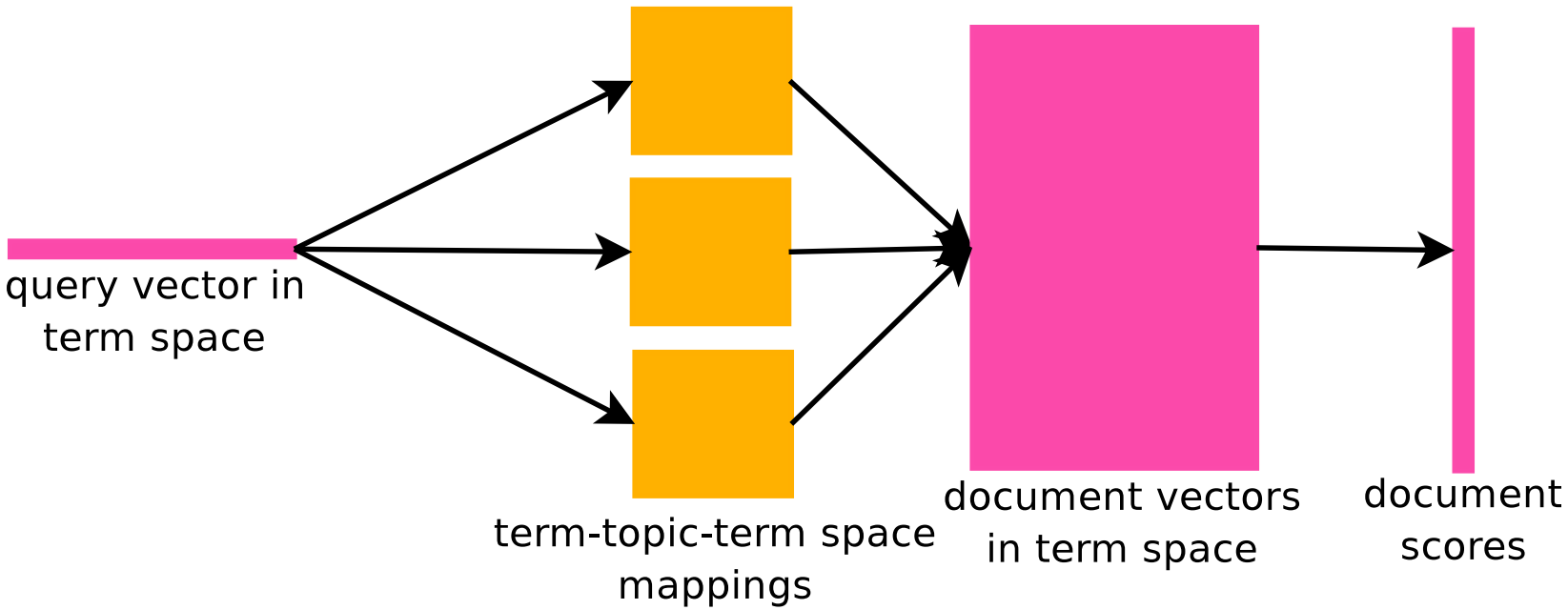


Storing the mapping

- The mapping is a $t \times t$ matrix (where t is the number of terms chosen).
- The size is independent of the number of eigenvalues chosen.
- The mapping is stored as in index for fast term look up.
- Accumulator is used to obtain top ranking terms for expanded query.
- Floating point values are quantised to 64 levels (6 bits).

Other possibilities

- Multiple query maps from different sources can assist in expanding polysemous terms.



Problems covered

Problems with precision:

- LSA on large document set produces poor precision.
- Synonymy is covered, polysemy is not.

Problems covered

Problems with precision:

- LSA on large document set produces poor precision.
- Synonymy is covered, polysemy is not.

Problems with query speed:

- Query in topic space has very few zero elements.
- Query process involves examining every term in the index.
- Document and query vectors in the topic space may have negative elements.
- Cannot use accumulator.

Problems covered

Problems with precision:

- LSA on large document set produces poor precision.
- Synonymy is covered, polysemy is not.

Problems with query speed:

- Query in topic space has very few zero elements.
- Query process involves examining every term in the index.
- Document and query vectors in the topic space may have negative elements.
- Cannot use accumulator.

Problems with document vector storage:

- Document vectors in the topic space have very few zero elements.
- Document vector elements are real values.
- Inverted index is useless.

Conclusion

- Latent semantic analysis (LSA) is a simple concept with firm mathematical foundations, but it has many problems.
- We proposed a query expansion which utilises LSA and overcomes most of the problems.