Vector Model Improvement by FCA and Topic Evolution

Petr Gajdoš Jan Martinovič

Department of Computer Science,
VŠB - Technical University of Ostrava,
tř. 17. listopadu 15, 708 33 Ostrava-Poruba
Czech Republic
Petr.Gajdos@vsb.cz
Jan.Martinovic@vsb.cz

April, 2005

Outline

Vector Model

- A query is represented by m dimensional vector

$$q = (q_1, q_2, \ldots, q_m),$$

  where $q_i \in \langle 0, 1 \rangle$.
- Each document d_i is represented by a vector

$$d_i = (w_{i1}, w_{i2}, \ldots, w_{im})$$

- An index file of the vector is represented by matrix, where
  - i-th row matches i-th document
  - j-th column matches j-th term

$$D = \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1m} \\ w_{21} & w_{22} & \ldots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \ldots & w_{nm} \end{pmatrix}$$

Vector Model

- A query is represented by m dimensional vector

$$q = (q_1, q_2, \ldots, q_m),$$

  where $q_i \in \langle 0, 1 \rangle$.
- Each document d_i is represented by a vector

$$d_i = (w_{i1}, w_{i2}, \ldots, w_{im})$$

- An index file of the vector is represented by matrix, where
  - i-th row matches i-th document
  - j-th column matches j-th term

$$D = \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1m} \\ w_{21} & w_{22} & \ldots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \ldots & w_{nm} \end{pmatrix}$$

Vector Model

- A query is represented by m dimensional vector

$$q = (q_1, q_2, \ldots, q_m),$$

  where $q_i \in \langle 0, 1 \rangle$.
- Each document d_i is represented by a vector

$$d_i = (w_{i1}, w_{i2}, \ldots, w_{im})$$

- An index file of the vector is represented by matrix, where
  - i-th row matches i-th document
  - j-th column matches j-th term

$$D = \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1m} \\ w_{21} & w_{22} & \ldots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \ldots & w_{nm} \end{pmatrix}$$

Vector Model

- **Coefficient of similarity** is a "distance" between the document's vector and the vector of the query
- **Cosine measure**:

$$sim(q, d_i) = \frac{\sum_{k=1}^{m} (q_k w_{ik})}{\sqrt{\sum_{k=1}^{m} (q_k)^2 \sum_{k=1}^{m} (w_{ik})^2}}$$

$$sim(d_i, d_j) = \frac{\sum_{k=1}^{m} (w_{ik} w_{jk})}{\sqrt{\sum_{k=1}^{m} (w_{ik})^2 \sum_{k=1}^{m} (w_{jk})^2}}$$

Formal Concept Analysis

- A **Formal context** $C := (G, M, I)$ consists of two sets G, M and one relation I between G and M.
  - elements of G are called objects
  - elements of M are called attributes

  If object $g \in G$ has an attribute $m \in M$, we write gIm or $(g, m) \in I$.
- The Incidence matrix

G \ M	m_1	m_2	...	m_l
g_1	0	1	...	1
g_2	1	0	...	1
...	...	...	...	...
g_k	1	1	...	0

Formal Concept Analysis

- For a set $A \subset G$ of objects we define

$$A^{\uparrow} = \{m \in M \mid gIm \text{ for all } g \in A\}$$

-the set of attributes common to the objects in A.

- Correspondingly, for a set $B \subset M$ of attributes we define

$$B^{\downarrow} = \{g \in G \mid gIm \text{ for all } m \in B\}$$

-the set of objects which have all attributes in B.

- A **formal concept** of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^{\uparrow} = B$ and $B^{\downarrow} = A$. We call A the extent and B the intent of the concept (A, B).

Formal Concept Analysis

- For a set $A \subset G$ of objects we define

$$A^{\uparrow} = \{m \in M \mid gIm \text{ for all } g \in A\}$$

-the set of attributes common to the objects in A.

- Correspondingly, for a set $B \subset M$ of attributes we define

$$B^{\downarrow} = \{g \in G \mid gIm \text{ for all } m \in B\}$$

-the set of objects which have all attributes in B.

- A **formal concept** of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^{\uparrow} = B$ and $B^{\downarrow} = A$. We call A the extent and B the intent of the concept (A, B).

Formal Concept Analysis

G \ M	m_1 1	m_2 1	m_3 1	m_4 1	m_5 1	m_6 1	m_7 1
g_1	x		x	x	x	x	
g_2		x	x				
g_3	x	x		x	x	x	x
g_4	x	x	x			x	

G \ M	m_1 1	m_2 1	m_3 1	m_4 1	m_5 1	m_6 1	m_7 1
g_1	x		x	x	x	x	
g_2		x	x				
g_3	x	x		x	x	x	x
g_4	x	x	x			x	

Diversity of object

$$do(g) = \sum_{m:m\in M \text{ and } (gIm)\in I} \lambda(m)$$

Sum of diversities of objects

$$sdo(C) = \sum_{g:g\in C} do(g)$$

Formal Concept Analysis

G \ M	m_1 1	m_2 1	m_3 1	m_4 1	m_5 1	m_6 1	m_7 1
g_1	x		x	x	x	x	
g_2		x	x				
g_3	x	x		x	x	x	x
g_4	x	x	x			x	

Diversity of concept

Let S is the set of objects of the concept C.

$$v(S) = \sum_{m \in M:(g,m) \in I \text{ for some } g \in S} \lambda(m)$$

It appears from Conjugate Moebius Function.

Formal Concept Analysis

The importance of selected object (document)

- Following formula has been obtained from observation and experiments

$$impo(g) = \sum_{C:C \ni g} \frac{sdo(C)}{v(S)} \lambda(A)\ do(g)$$

where S is the set of objects and A is the set of attributes of the concept C.

- $\frac{sdo(C)}{v(S)}$ The range of covered attributes (words).
  It depends on weights of attributes and differences between objects of selected concept.
- $\lambda(A)$ The weight of unique attributes.
- $do(g)$ The weight of attributes owned by object (document).
  This is used for objects' differentiation in the same concept.

Formal Concept Analysis

The importance of selected object (document)

- Following formula has been obtained from observation and experiments

$$impo(g) = \sum_{C:C \ni g} \frac{sdo(C)}{v(S)} \lambda(A)\ do(g)$$

where S is the set of objects and A is the set of attributes of the concept C.

- $\frac{sdo(C)}{v(S)}$ The range of covered attributes (words).
  It depends on weights of attributes and differences between objects of selected concept.
- $\lambda(A)$ The weight of unique attributes.
- $do(g)$ The weight of attributes owned by object (document).
  This is used for objects' differentiation in the same concept.

Formal Concept Analysis

The importance of selected object (document)

- Following formula has been obtained from observation and experiments

$$impo(g) = \sum_{C:C \ni g} \frac{sdo(C)}{v(S)} \lambda(A)\ do(g)$$

where S is the set of objects and A is the set of attributes of the concept C.

- $\frac{sdo(C)}{v(S)}$ The range of covered attributes (words).
  It depends on weights of attributes and differences between objects of selected concept.
- $\lambda(A)$ The weight of unique attributes.
- $do(g)$ The weight of attributes owned by object (document).
  This is used for objects' differentiation in the same concept.

Formal Concept Analysis

The importance of selected object (document)

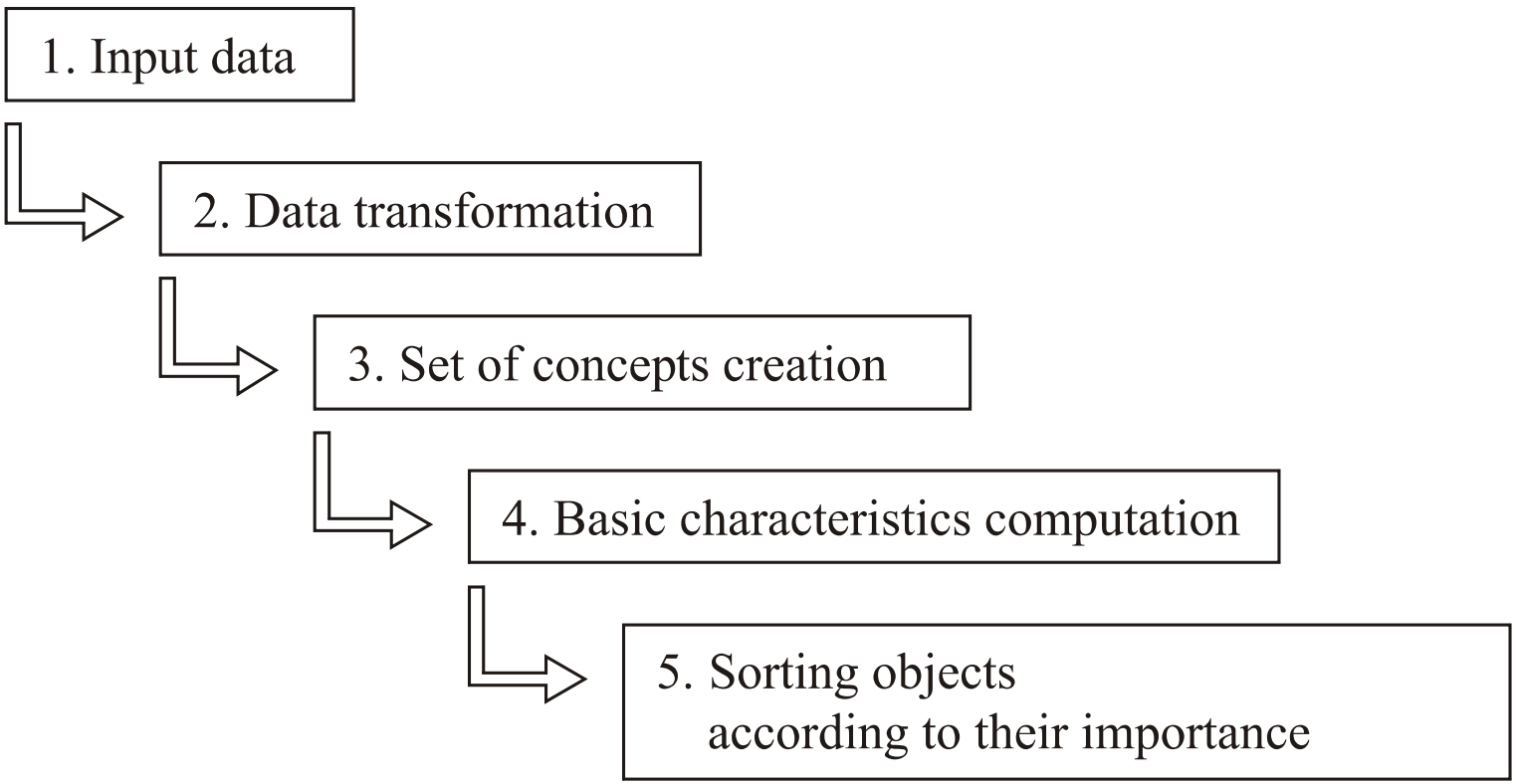
- Following formula has been obtained from observation and experiments

$$impo(g) = \sum_{C:C \ni g} \frac{sdo(C)}{v(S)} \lambda(A)\ do(g)$$

where S is the set of objects and A is the set of attributes of the concept C.

- $\frac{sdo(C)}{v(S)}$ The range of covered attributes (words).
  It depends on weights of attributes and differences between objects of selected concept.
- $\lambda(A)$ The weight of unique attributes.
- $do(g)$ The weight of attributes owned by object (document).
  This is used for objects' differentiation in the same concept.

Obtaining the importances of documents by FCA



Topic Evolution

- Evolution of topic
  - documents may use different words to describe the same theme
  - list of documents related to theme, which is described by query
  - result of query
  - a query may consists of whole document.
- Clusters generation
  - TOPIC-CA algorithm
  - TOPIC-FCA algorithm
- Reordering algorithm
  - SORT-EACH alg. moves all documents in a result of the vector model query so that the documents belonging to the same evolution of topic are closer to each other. It calls CA or FCA Topic algorithm.

Topic Evolution

- Evolution of topic
  - documents may use different words to describe the same theme
  - list of documents related to theme, which is described by query
  - result of query
  - a query may consists of whole document.
- Clusters generation
  - TOPIC-CA algorithm
  - TOPIC-FCA algorithm
- Reordering algorithm
  - SORT-EACH alg. moves all documents in a result of the vector model query so that the documents belonging to the same evolution of topic are closer to each other. It calls CA or FCA Topic algorithm.

Topic Evolution

- Evolution of topic
  - documents may use different words to describe the same theme
  - list of documents related to theme, which is described by query
  - result of query
  - a query may consists of whole document.
- Clusters generation
  - TOPIC-CA algorithm
  - TOPIC-FCA algorithm
- Reordering algorithm
  - SORT-EACH alg. moves all documents in a result of the vector model query so that the documents belonging to the same evolution of topic are closer to each other. It calls CA or FCA Topic algorithm.

Topic Evolution

All clusters of documents have been created using Cosine measure. Also we have now the hierarchy of documents (dendogram).

TOPIC-CA algorithm

1. Next we choose the total number of documents in each topic ('level').
2. Then we find leaf cluster which contains selected relevant document.
3. We pass through the hierarchy.
4. We explore neighbouring clusters. First we select the cluster created on the highest sub-level. Each document, which we find, we add to the result list. When the count of all documents in the result list equals to 'level' we break finding.
5. Go to the step 3 (we are going to compute Topic Evolution for next document).

Topic Evolution

All clusters of documents have been created using Cosine measure. Also we have now the hierarchy of documents (dendogram).

TOPIC-CA algorithm

1. Next we choose the total number of documents in each topic ('level').
2. Then we find leaf cluster which contains selected relevant document.
3. We pass through the hierarchy.
4. We explore neighbouring clusters. First we select the cluster created on the highest sub-level. Each document, which we find, we add to the result list. When the count of all documents in the result list equals to 'level' we break finding.
5. Go to the step 3 (we are going to compute Topic Evolution for next document).

Topic Evolution

All clusters of documents have been created using Cosine measure. Also we have now the hierarchy of documents (dendogram).

TOPIC-CA algorithm

1. Next we choose the total number of documents in each topic ('level').
2. Then we find leaf cluster which contains selected relevant document.
3. We pass through the hierarchy.
4. We explore neighbouring clusters. First we select the cluster created on the highest sub-level. Each document, which we find, we add to the result list. When the count of all documents in the result list equals to 'level' we break finding.
5. Go to the step 3 (we are going to compute Topic Evolution for next document).

Topic Evolution

All clusters of documents have been created using Cosine measure. Also we have now the hierarchy of documents (dendogram).

TOPIC-CA algorithm

1. Next we choose the total number of documents in each topic ('level').
2. Then we find leaf cluster which contains selected relevant document.
3. We pass through the hierarchy.
4. We explore neighbouring clusters. First we select the cluster created on the highest sub-level. Each document, which we find, we add to the result list. When the count of all documents in the result list equals to 'level' we break finding.
5. Go to the step 3 (we are going to compute Topic Evolution for next document).

Topic Evolution

All clusters of documents have been created using Cosine measure. Also we have now the hierarchy of documents (dendogram).

TOPIC-CA algorithm

1. Next we choose the total number of documents in each topic ('level').
2. Then we find leaf cluster which contains selected relevant document.
3. We pass through the hierarchy.
4. We explore neighbouring clusters. First we select the cluster created on the highest sub-level. Each document, which we find, we add to the result list. When the count of all documents in the result list equals to 'level' we break finding.
5. Go to the step 3 (we are going to compute Topic Evolution for next document).

Topic Evolution

All concepts have been already computed.

TOPIC-FCA algorithm

1. We make the query transformation. It means that we create weighted vector of terms.
2. We compute the importances of documents (objects) and we make the list of the documents and their importances.
3. We find the relevant document rel_d in the ordered list.
4. In finite steps, we look for "nearest" documents. The "nearest" document is the document, that has the smallest difference between its weight and the weight of rel_d. Founded document is excluded before repeating of this step.

Topic Evolution

All concepts have been already computed.

TOPIC-FCA algorithm

1. We make the query transformation. It means that we create weighted vector of terms.
2. We compute the importances of documents (objects) and we make the list of the documents and their importances.
3. We find the relevant document rel_d in the ordered list.
4. In finite steps, we look for "nearest" documents. The "nearest" document is the document, that has the smallest difference between its weight and the weight of rel_d. Founded document is excluded before repeating of this step.

Topic Evolution

All concepts have been already computed.

TOPIC-FCA algorithm

1. We make the query transformation. It means that we create weighted vector of terms.
2. We compute the importances of documents (objects) and we make the list of the documents and their importances.
3. We find the relevant document rel_d in the ordered list.
4. In finite steps, we look for "nearest" documents. The "nearest" document is the document, that has the smallest difference between its weight and the weight of rel_d. Founded document is excluded before repeating of this step.

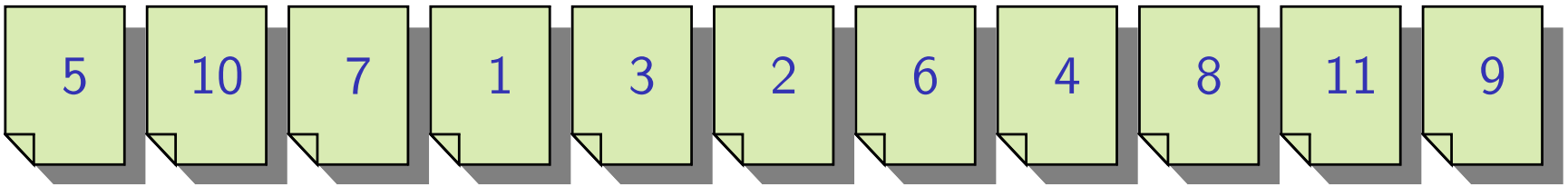
Topic Evolution

All concepts have been already computed.

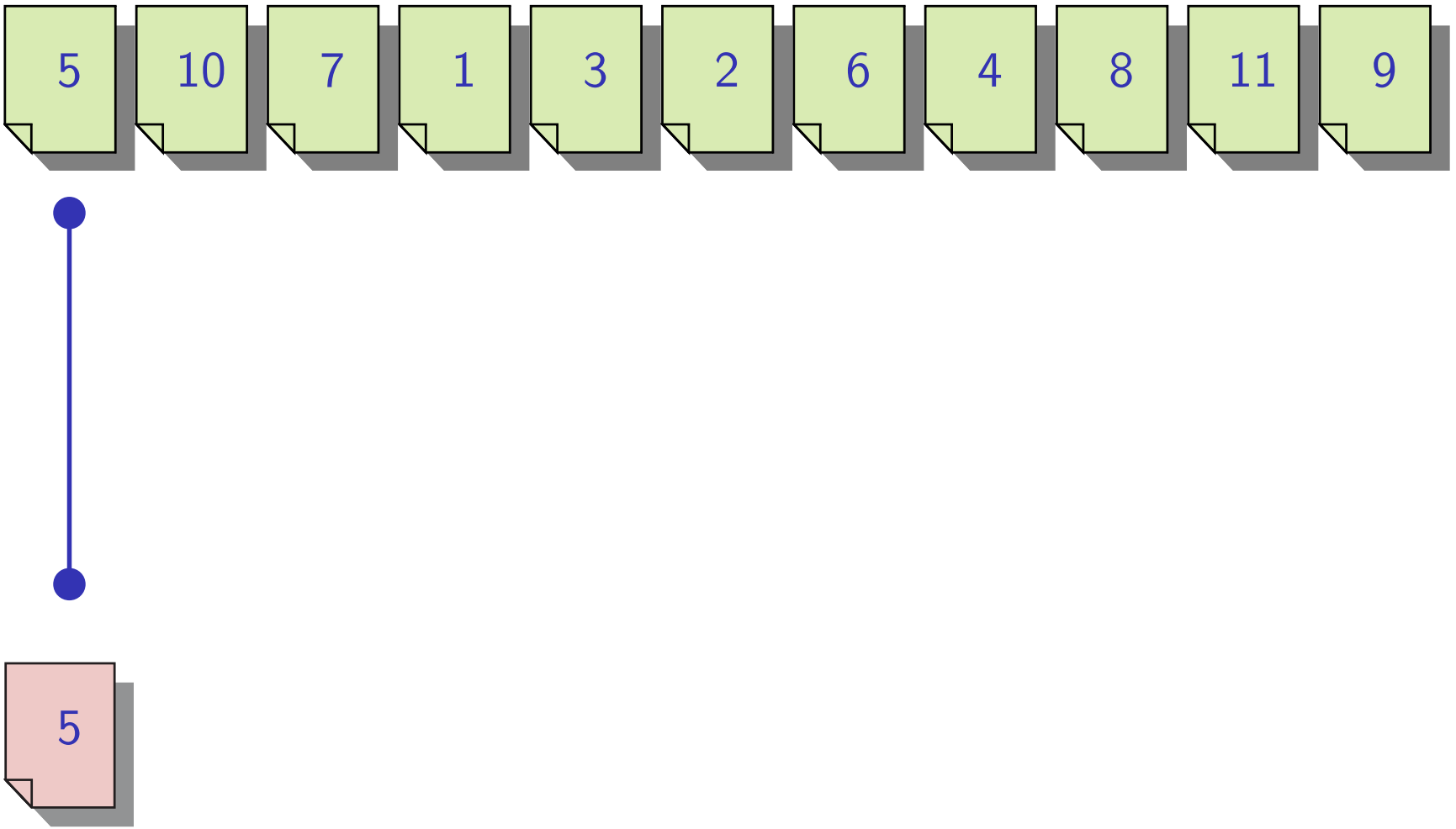
TOPIC-FCA algorithm

1. We make the query transformation. It means that we create weighted vector of terms.
2. We compute the importances of documents (objects) and we make the list of the documents and their importances.
3. We find the relevant document rel_d in the ordered list.
4. In finite steps, we look for "nearest" documents. The "nearest" document is the document, that has the smallest difference between its weight and the weight of rel_d. Founded document is excluded before repeating of this step.

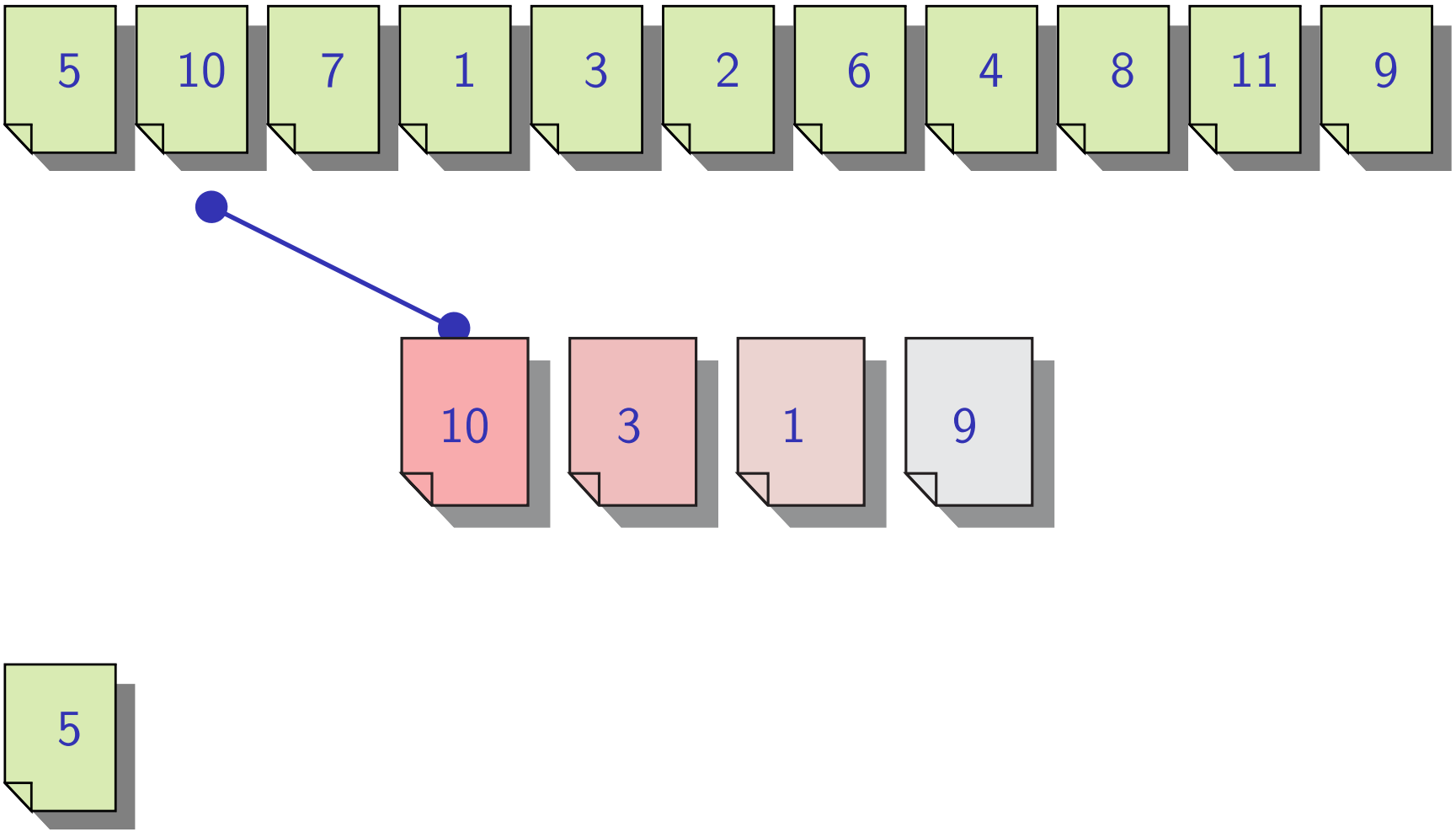
Topic Evolution - SORT-EACH algorithm



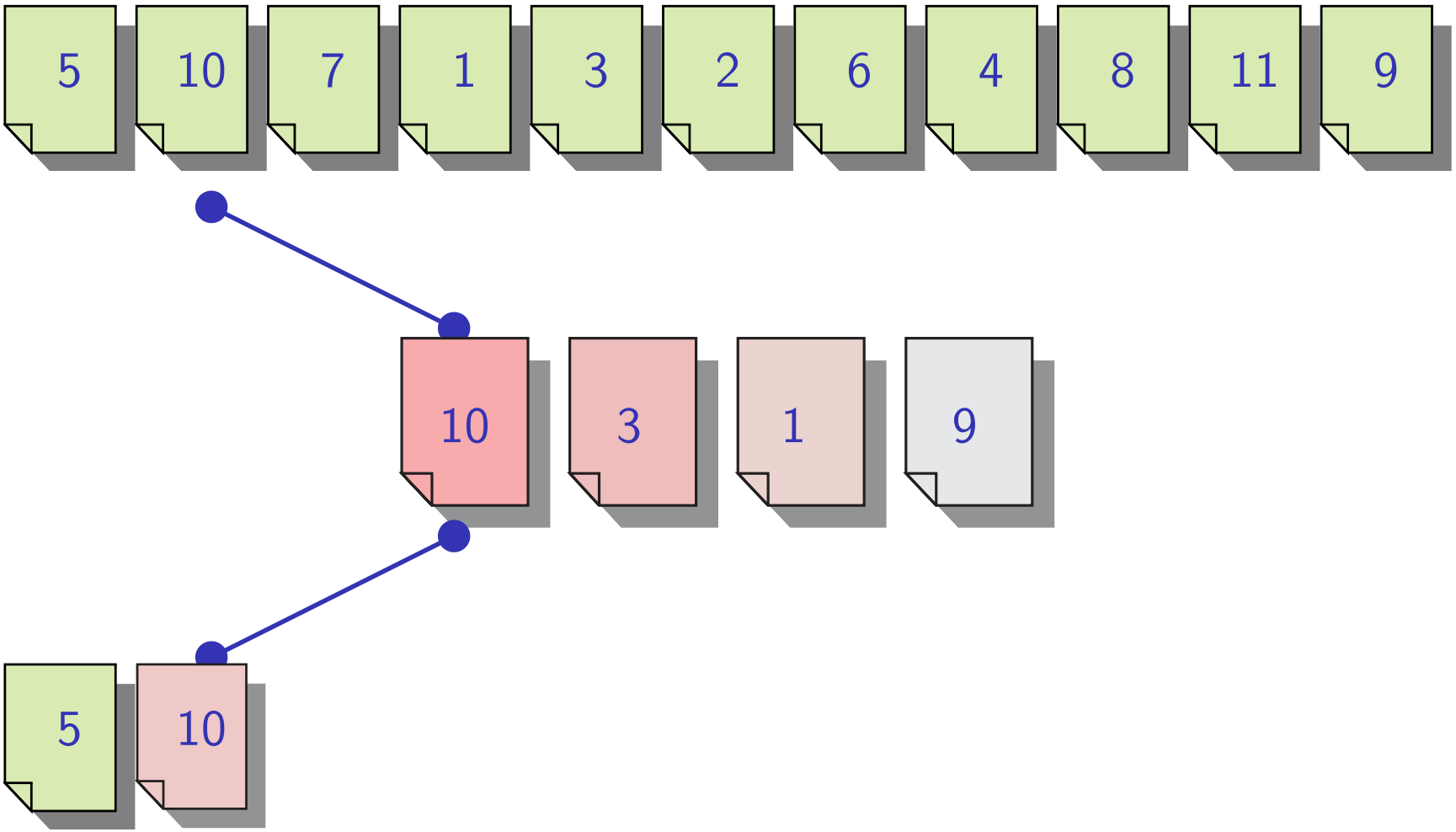
Topic Evolution - SORT-EACH algorithm



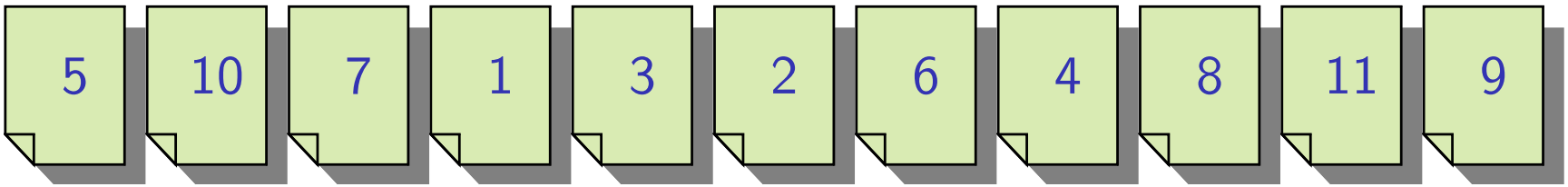
Topic Evolution - SORT-EACH algorithm

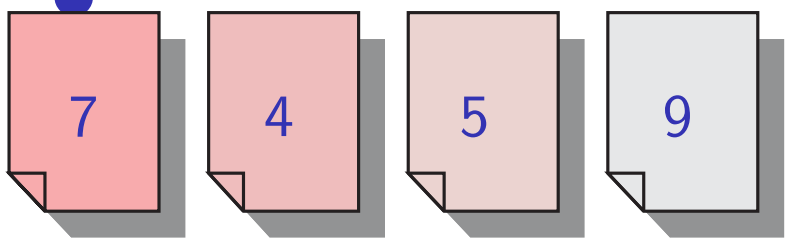


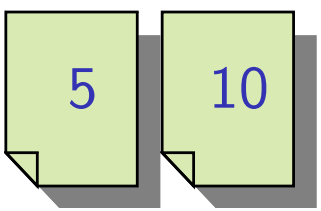
Topic Evolution - SORT-EACH algorithm



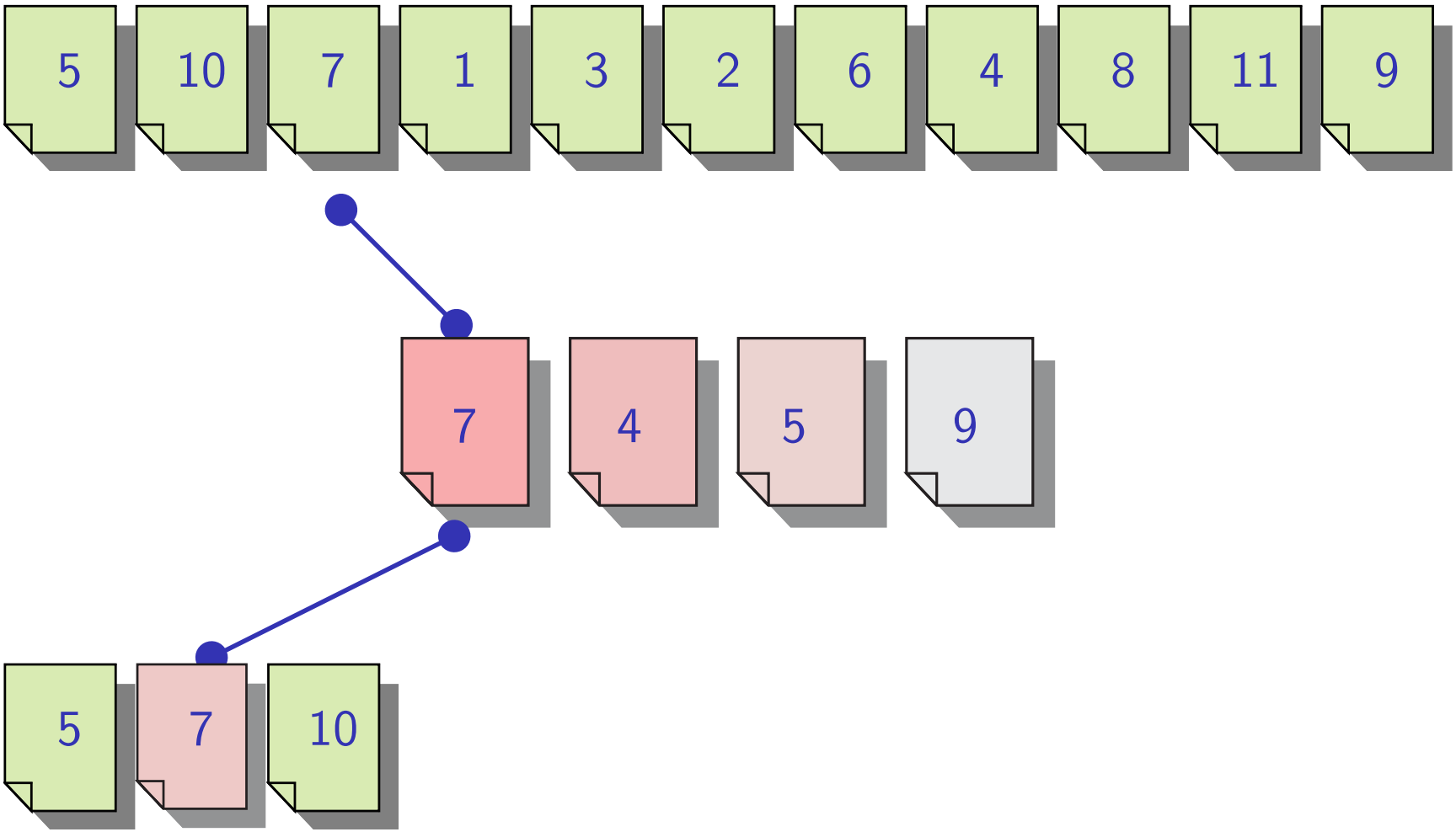
Topic Evolution - SORT-EACH algorithm



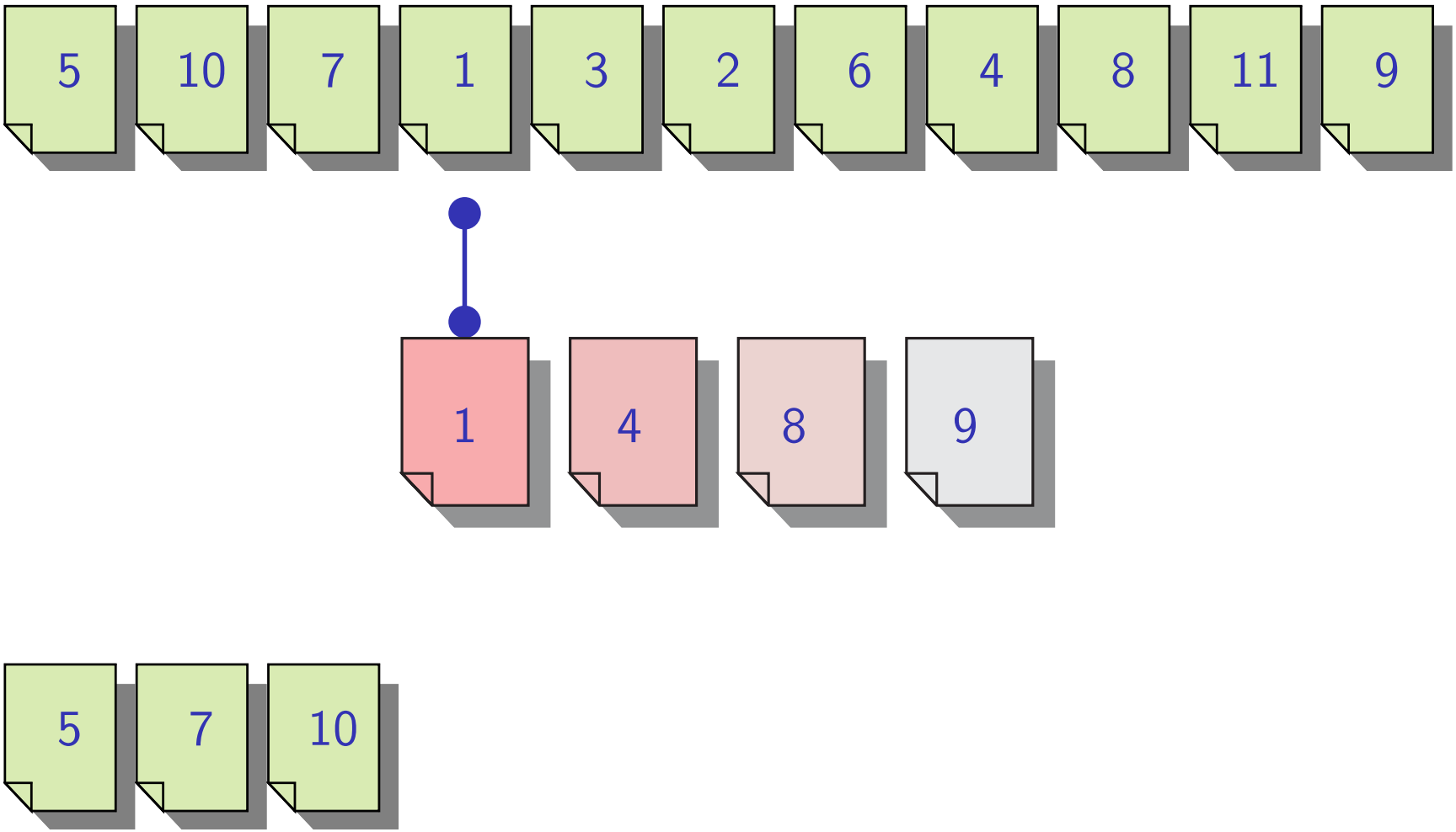




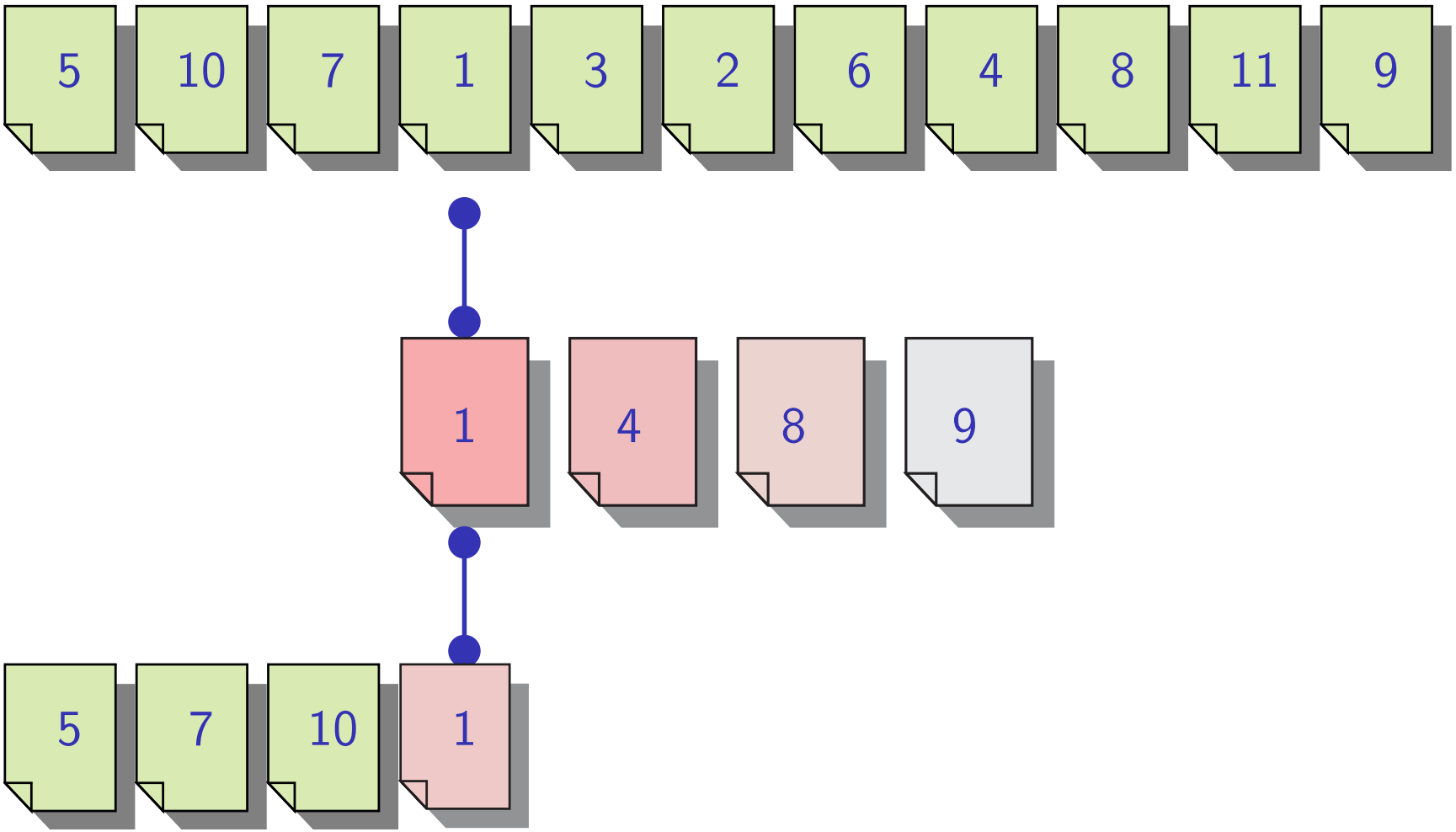
Topic Evolution - SORT-EACH algorithm



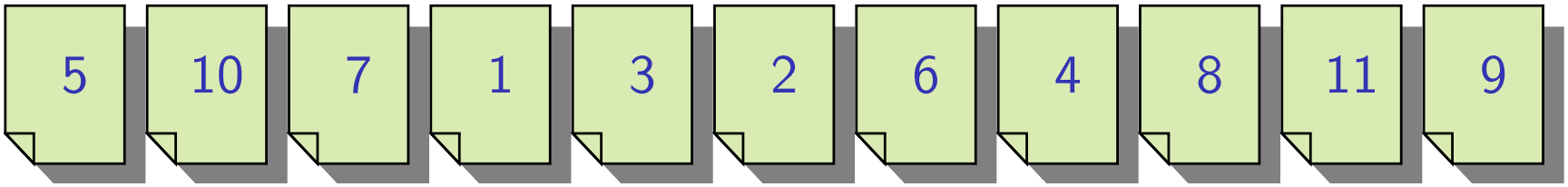
Topic Evolution - SORT-EACH algorithm

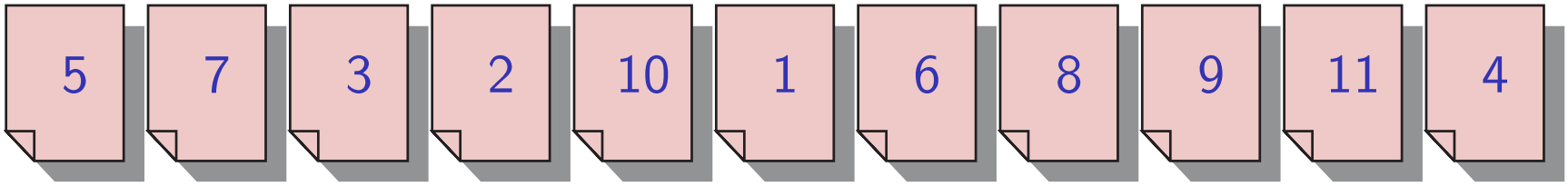


Topic Evolution - SORT-EACH algorithm



Topic Evolution - SORT-EACH algorithm





Illustrative samples

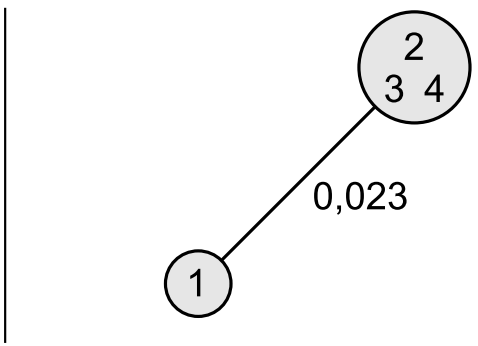
query	1 t_1	1 t_2	1 t_3	1 t_4	1 t_5	1 t_4	1 t_7	1 t_8	1 t_9	1 t_{10}	1 t_{11}	1 t_{12}	Document's importance	Vector query
doc. 1	1	1	1		1								66.66666667	0.57735
doc. 2				**1**	**1**	**1**							**38**	**0.5**
doc. 3							1	1	1				36	0.5
doc. 4										1	1	1	36	0.5

Table: The results after inserted query "11111111111"

Illustrative samples

query	1 1 1 1 1 1 1 1 1 1 1 1 t_1 t_2 t_3 t_4 t_5 t_4 t_7 t_8 t_9 t_{10} t_{11} t_{12}	Document's importance	Vector query
doc. 1	1 1 1 1	66.66666667	0.57735
doc. 2	**1 1 1**	**38**	**0.5**
doc. 3	1 1 1	36	0.5
doc. 4	1 1 1	36	0.5

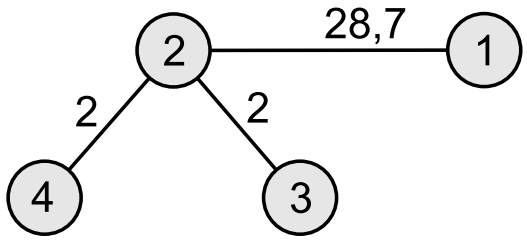
Table: The results after inserted query "111111111111"

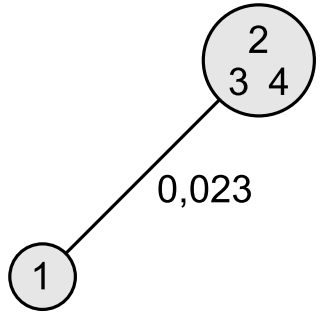


Illustrative samples

query	1 1 1 1 1 1 1 1 1 1 1 1 t_1 t_2 t_3 t_4 t_5 t_4 t_7 t_8 t_9 t_{10} t_{11} t_{12}	Document's importance	Vector query
doc. 1	1 1 1 1	66.66666667	0.57735
doc. 2	**1 1 1**	**38**	**0.5**
doc. 3	1 1 1	36	0.5
doc. 4	1 1 1	36	0.5

Table: The results after inserted query "11111111111"





Illustrative samples

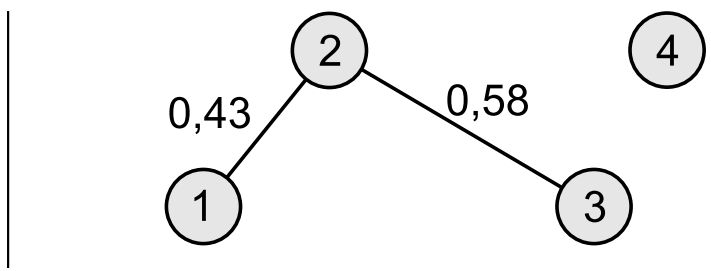
query	0	0	0	1	1	1	0	0	0	0	0	0	Document's importance	Vector query
	t_1	t_2	t_3	t_4	t_5	t_4	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}		
doc. 1				1	1			1	1	1	1	1	94.93333333	0.436436
doc. 2				**1**	**1**	**1**			**1**				**53.2**	**0.866025**
doc. 3						1	1	1	1				47	0.288675
doc. 4									1	1	1	1	26	0

Table: The results after inserted query "000111000000"

Illustrative samples

query	0 0 0 1 1 1 0 0 0 0 0 0	Document's importance	Vector query
	t_1 t_2 t_3 t_4 t_5 t_4 t_7 t_8 t_9 t_{10} t_{11} t_{12}		
doc. 1	1 1 1 1 1 1	94.93333333	0.436436
doc. 2	**1 1 1 1**	**53.2**	**0.866025**
doc. 3	1 1 1 1	47	0.288675
doc. 4	1 1 1 1	26	0

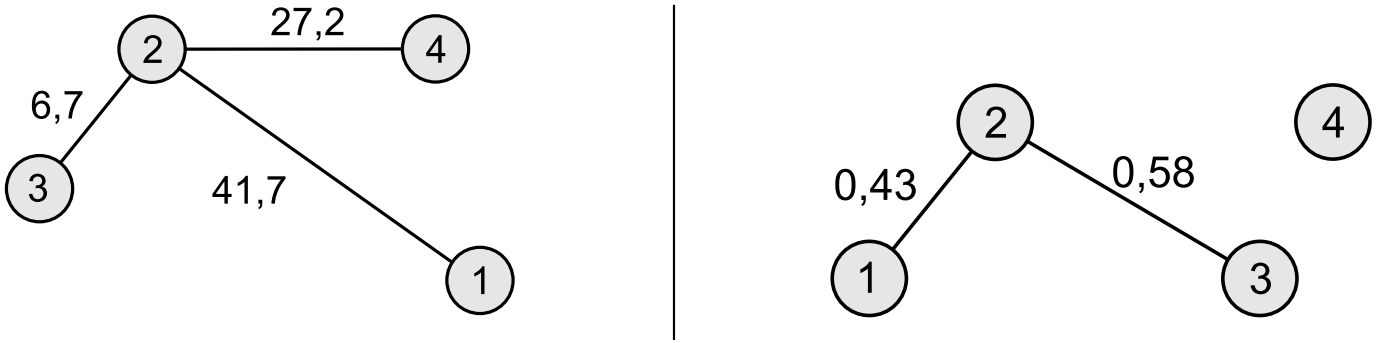
Table: The results after inserted query "000111000000"



Illustrative samples

query	0 0 0 1 1 1 0 0 0 0 0 0												Document's importance	Vector query
	t_1	t_2	t_3	t_4	t_5	t_4	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}		
doc. 1				1	1			1	1	1	1	1	94.93333333	0.436436
doc. 2				**1**	**1**	**1**			**1**				**53.2**	**0.866025**
doc. 3						1	1	1	1				47	0.288675
doc. 4									1	1	1	1	26	0

Table: The results after inserted query "000111000000"



Illustrative samples

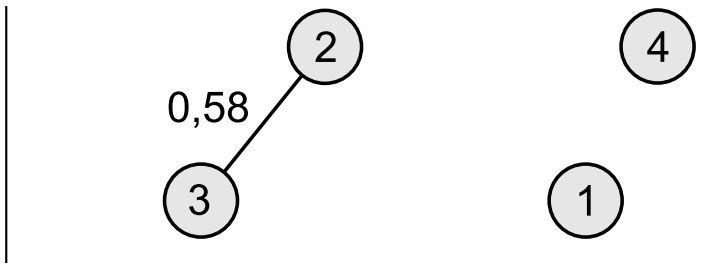
query	0	0	0	1	1	1	0	0	0	0	0	0	Document's importance	Vector query
	t_1	t_2	t_3	t_4	t_5	t_4	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}		
doc. 1								1	1	1	1	1	41.86111111	0
doc. 2				**1**	**1**	**1**			**1**				**44.5**	**0.866025**
doc. 3						1	1	1	1				45.83333333	0.288675
doc. 4									1	1	1	1	28.6	0

Table: The results after inserted query "000111000000"

Illustrative samples

query	0 0 0 1 1 1 0 0 0 0 0 0	Document's importance	Vector query
	t_1 t_2 t_3 t_4 t_5 t_4 t_7 t_8 t_9 t_{10} t_{11} t_{12}		
doc. 1	1 1 1 1 1	41.86111111	0
doc. 2	**1 1 1 1**	**44.5**	**0.866025**
doc. 3	1 1 1 1	45.83333333	0.288675
doc. 4	1 1 1 1	28.6	0

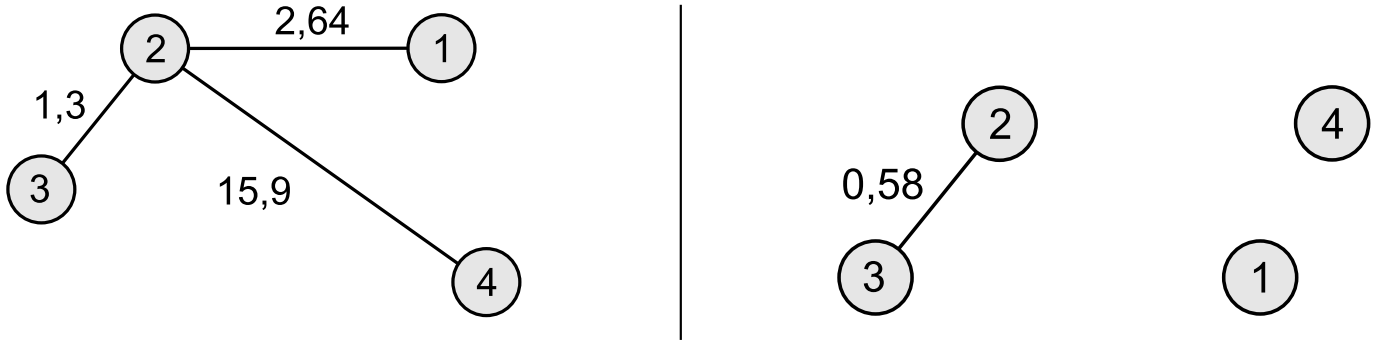
Table: The results after inserted query "000111000000"



Illustrative samples

query	0 0 0 1 1 1 0 0 0 0 0 0	Document's importance	Vector query
	t_1 t_2 t_3 t_4 t_5 t_4 t_7 t_8 t_9 t_{10} t_{11} t_{12}		
doc. 1	1 1 1 1 1	41.86111111	0
doc. 2	**1 1 1 1**	**44.5**	**0.866025**
doc. 3	1 1 1 1	45.83333333	0.288675
doc. 4	1 1 1 1	28.6	0

Table: The results after inserted query "000111000000"



Conclusion

- We have described new method for vector query improvement based on formal concept analysis and Moebius inverse function.
- The known deficiencies of vector model have been suppressed using TOPICs and SEARCH-EACH algorithms.
- Our presented methods can be applied on small data sets or on large collections of documents.

Future Work

- test our method on large data collections
- improve all algorithms by usage sparse matrix based on finite automata
- usage this method for collection preprocessing according to specific dictionaries (mathematic, medicine, ...)

References I


Berry, M. W (Ed.)
Survey of Text Mining: Clustering Classification, and Retrieval. Springer Verlag 2003.


Ganter B., Wille R.
Formal Concept Analysis. Springer-Verlag, Berlin, Heidelberg, 1999.


C.J. van Rijsbergen
Information Retrieval (second ed.). London, Butterworths, 1979.


Ďuráková, D., Gajdoš, P.
Indicators Valuation using FCA and Moebius Inversion Function. DATAKON, Brno, 2004, IBSN 80-210-3516-1.

References II


Dvorský J., Martinovič J., Pokorný J., Snášel V.
A Search topics in Collection of Documents.(in Czech). Znalosti 2004, ISBN: 80-248-0456-5.


Keith Van Rijsbergen
The Geometry of Information Retrieva. Cambridge University Press, 2004.


Kummamuru K, Lotlikar R., Roy S., Singal K., Krishnapuram R.
A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. WWW2004, New York, USA.


Nehring, K.
A Theory of Diversity. Ecometrica 70, 1155-1198, 2002.

Thank you for your attention.