LEMP: Fast Retrieval of Large Entries in a Matrix Product

Christina Teflioudi[1] Rainer Gemulla[2] Olga Mykytiuk

[1]Max Planck Institute for Computer Science
Saarbrücken, Germany

[2]Mannheim University
Mannheim, Germany

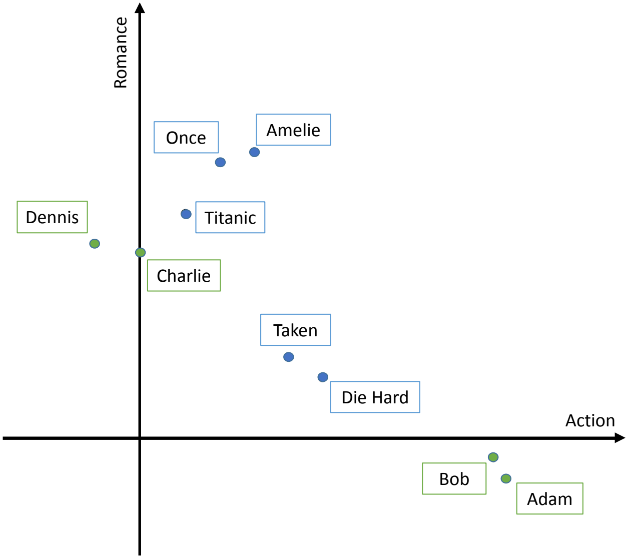
May 4, 2015

Recommender Systems

- Problem
  - Set of users
  - Set of items (movies, books, jokes, products, stories, ...)
  - Feedback (ratings, purchase, click-through, tags, ...)
- Predict the preference of each user for each item

$$
\begin{array}{l c}
 & \begin{array}{ccccc} \textit{Die Hard} & \textit{Taken} & \textit{Once} & \textit{Amelie} & \textit{Titanic} \end{array} \\
\begin{array}{l} \textit{Adam} \\ \textit{Bob} \\ \textit{Charlie} \\ \textit{Dennis} \end{array} &
\begin{pmatrix}
5 & ? & 1 & 2 & ? \\
5 & 4 & ? & ? & 1 \\
2 & ? & 5 & ? & 4 \\
? & 1 & 5 & 5 & ?
\end{pmatrix}
\end{array}
$$

Latent Factor Models



Recommender Systems

- Given
  - A query matrix $\mathbf{Q}$

$$
\begin{array}{l}
Adam \\
Bob \\
Charlie \\
Dennis
\end{array}
\underbrace{\begin{pmatrix}
3.2 & -0.4 \\
3.1 & -0.2 \\
0 & 1.8 \\
-0.4 & 1.9
\end{pmatrix}}_{\mathbf{Q}^T}
$$

Recommender Systems

- Given
  - A query matrix $\mathbf{Q}$
  - A probe matrix $\mathbf{P}$

$$\begin{array}{ccccc} \textit{Die Hard} & \textit{Taken} & \textit{Once} & \textit{Amelie} & \textit{Titanic} \end{array}$$

$$\begin{pmatrix} 1.6 & 1.3 & 0.7 & 1 & 0.4 \\ 0.6 & 0.8 & 2.7 & 2.8 & 2.2 \end{pmatrix} \mathbf{P}$$

$$\begin{array}{l} \textit{Adam} \\ \textit{Bob} \\ \textit{Charlie} \\ \textit{Dennis} \end{array} \underbrace{\begin{pmatrix} 3.2 & -0.4 \\ 3.1 & -0.2 \\ 0 & 1.8 \\ -0.4 & 1.9 \end{pmatrix}}_{\mathbf{Q}^T}$$

Recommender Systems

- Given
  - A query matrix $\mathbf{Q}$
  - A probe matrix $\mathbf{P}$
  - A threshold $\theta > 0$
- Find good recommendations

$$\begin{array}{c} \begin{array}{ccccc} \textit{Die Hard} & \textit{Taken} & \textit{Once} & \textit{Amelie} & \textit{Titanic} \end{array} \\ \begin{pmatrix} 1.6 & 1.3 & 0.7 & 1 & 0.4 \\ 0.6 & 0.8 & 2.7 & 2.8 & 2.2 \end{pmatrix} \mathbf{P} \end{array}$$

$$\begin{array}{l} \textit{Adam} \\ \textit{Bob} \\ \textit{Charlie} \\ \textit{Dennis} \end{array} \begin{pmatrix} 3.2 & -0.4 \\ 3.1 & -0.2 \\ 0 & 1.8 \\ -0.4 & 1.9 \end{pmatrix}$$

$$\mathbf{Q}^T$$

Recommender Systems

- Given
  - A query matrix $\mathbf{Q}$
  - A probe matrix $\mathbf{P}$
  - A threshold $\theta > 0$
- Find good recommendations
  - All entries in $\mathbf{Q}^T\mathbf{P}$ that are $\geq \theta$

$$\begin{array}{ccccc} \textit{Die Hard} & \textit{Taken} & \textit{Once} & \textit{Amelie} & \textit{Titanic} \end{array}$$

$$\begin{pmatrix} 1.6 & 1.3 & 0.7 & 1 & 0.4 \\ 0.6 & 0.8 & 2.7 & 2.8 & 2.2 \end{pmatrix} \mathbf{P}$$

$$\begin{array}{l} \textit{Adam} \\ \textit{Bob} \\ \textit{Charlie} \\ \textit{Dennis} \end{array} \underbrace{\begin{pmatrix} 3.2 & -0.4 \\ 3.1 & -0.2 \\ 0 & 1.8 \\ -0.4 & 1.9 \end{pmatrix}}_{\mathbf{Q}^T} \quad \underbrace{\begin{pmatrix} \mathbf{4.9} & \mathbf{3.8} & 1.2 & 2.1 & 0.4 \\ \mathbf{4.8} & \mathbf{3.9} & 1.6 & 2.5 & 0.8 \\ 1 & 1.4 & \mathbf{4.9} & \mathbf{5.0} & \mathbf{4.0} \\ 0.5 & 1 & \mathbf{4.9} & \mathbf{4.9} & \mathbf{4.0} \end{pmatrix}}_{\mathbf{Q}^T\mathbf{P}}$$

Recommender Systems

- Given
  - A query matrix $\mathbf{Q}$
  - A probe matrix $\mathbf{P}$
  - A threshold $\theta > 0$
- Find good recommendations
  - All entries in $\mathbf{Q}^T\mathbf{P}$ that are $\geq \theta$
  - Each entry is an inner product $\mathbf{q}^T\mathbf{p} = \sum_{i=1}^{r} q_i p_i$

$$
\begin{array}{ccccc}
\textit{Die Hard} & \textit{Taken} & \textit{Once} & \textit{Amelie} & \textit{Titanic}
\end{array}
$$

$$
\begin{pmatrix} 1.6 & 1.3 & 0.7 & 1 & 0.4 \\ 0.6 & 0.8 & 2.7 & 2.8 & 2.2 \end{pmatrix} \mathbf{P}
$$

$$
\begin{array}{l}
\textit{Adam} \\ \textit{Bob} \\ \textit{Charlie} \\ \textit{Dennis}
\end{array}
\underbrace{\begin{pmatrix} 3.2 & -0.4 \\ 3.1 & -0.2 \\ 0 & 1.8 \\ -0.4 & 1.9 \end{pmatrix}}_{\mathbf{Q}^T}
\qquad
\underbrace{\begin{pmatrix} \mathbf{4.9} & \mathbf{3.8} & 1.2 & 2.1 & 0.4 \\ \mathbf{4.8} & \mathbf{3.9} & 1.6 & 2.5 & 0.8 \\ 1 & 1.4 & \mathbf{4.9} & \mathbf{5.0} & \mathbf{4.0} \\ 0.5 & 1 & \mathbf{4.9} & \mathbf{4.9} & \mathbf{4.0} \end{pmatrix}}_{\mathbf{Q}^T\mathbf{P}}
$$

Problem Statement

Maximum Inner Product Search

Find pairs of vectors with large inner products

Given

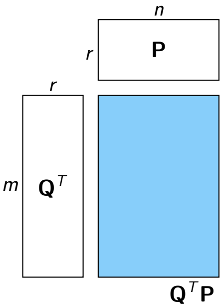
- a **query vector** $\mathbf{q}^T$ (from matrix $\mathbf{Q}^T$)
- a set $\mathbf{P}$ of **probe vectors** (the matrix $\mathbf{P}$)
- a threshold $\theta > 0$

Find

- all vectors $\mathbf{p}$ such that $\mathbf{q}^T\mathbf{p} \geq \theta$

Naive Solution

- Compute full matrix product $\mathbf{Q}^T\mathbf{P}$
- Determine which entries are $\geq \theta$
- Complexity $O(mnr)$
- Usually
  - m, n : order of millions
  - $10 < \mathrm{r} < 500$
- Example
  - m $=$ 10 millions
  - n $=$ 1 million
  - #entries $=$ 10 trillion
  - Avg. Inner Product Time $=$ 100nsec
  - Runtime $>$ 11 days
- Can we do better than that?

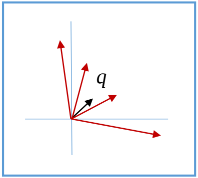


When is an inner product large?

$$\mathbf{q}^T\mathbf{p} = \|\mathbf{q}\|\|\mathbf{p}\| \cos\angle(\mathbf{q},\mathbf{p}) \geq \theta \ , \ -1 \leq \cos\angle(\mathbf{q},\mathbf{p}) \leq 1$$

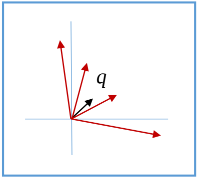
When is an inner product large?

$$\mathbf{q}^T\mathbf{p} = \|\mathbf{q}\|\|\mathbf{p}\| \cos\angle(\mathbf{q},\mathbf{p}) \geq \theta \text{ , } -1 \leq \cos\angle(\mathbf{q},\mathbf{p}) \leq 1$$

	short	**medium**	**long**
Length ($\|\mathbf{q}\|\|\mathbf{p}\|$)	$(< \theta)$	$(\approx \theta)$	$(\gg \theta)$
			
Angle ($\cos\angle(\mathbf{q},\mathbf{p})$)	small positive large positive		
Suitable method			

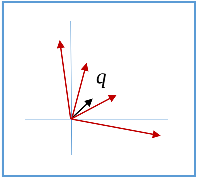
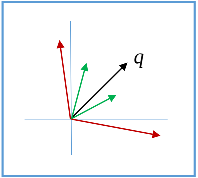
When is an inner product large?

$$\mathbf{q}^T\mathbf{p} = \|\mathbf{q}\|\|\mathbf{p}\| \cos\angle(\mathbf{q},\mathbf{p}) \geq \theta \ , \ -1 \leq \cos\angle(\mathbf{q},\mathbf{p}) \leq 1$$

Length ($\|\mathbf{q}\|\|\mathbf{p}\|$)	**short** ($< \theta$)	**medium** ($\approx \theta$)	**long** ($\gg \theta$)
			
Angle ($\cos\angle(\mathbf{q},\mathbf{p})$)	small positive large positive		
Suitable method	Prune		

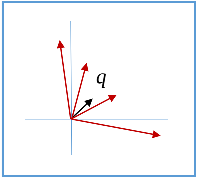
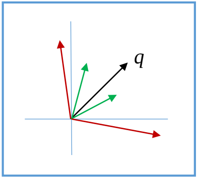
When is an inner product large?

$$\mathbf{q}^T\mathbf{p} = \lVert\mathbf{q}\rVert\lVert\mathbf{p}\rVert \cos\angle(\mathbf{q},\mathbf{p}) \geq \theta \ , \ -1 \leq \cos\angle(\mathbf{q},\mathbf{p}) \leq 1$$

Length ($\lVert\mathbf{q}\rVert\lVert\mathbf{p}\rVert$)	**short** ($< \theta$)	**medium** ($\approx \theta$)	**long** ($\gg \theta$)
			
Angle ($\cos\angle(\mathbf{q},\mathbf{p})$)	small positive large positive	small positive large positive	
Suitable method	Prune		

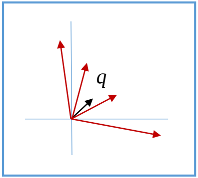
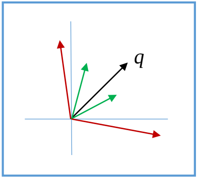
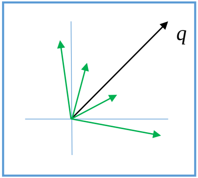
When is an inner product large?

$$\mathbf{q}^T\mathbf{p} = \|\mathbf{q}\|\|\mathbf{p}\| \cos\angle(\mathbf{q},\mathbf{p}) \geq \theta \ , \ -1 \leq \cos\angle(\mathbf{q},\mathbf{p}) \leq 1$$

Length ($\|\mathbf{q}\|\|\mathbf{p}\|$)	**short** ($< \theta$)	**medium** ($\approx \theta$)	**long** ($\gg \theta$)
			
Angle ($\cos\angle(\mathbf{q},\mathbf{p})$)	small positive large positive	small positive large positive	
Suitable method	Prune	Cosine Similarity Search $\theta' = \frac{\theta}{\|\mathbf{q}\|\|\mathbf{p}\|}$	

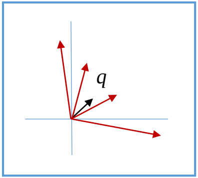
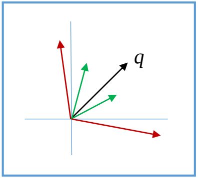
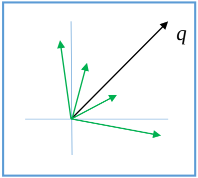
When is an inner product large?

$$\mathbf{q}^T\mathbf{p} = \|\mathbf{q}\|\|\mathbf{p}\|\cos\angle(\mathbf{q},\mathbf{p}) \geq \theta\ ,\ -1 \leq \cos\angle(\mathbf{q},\mathbf{p}) \leq 1$$

Length ($\|\mathbf{q}\|\|\mathbf{p}\|$)	**short** ($< \theta$)	**medium** ($\approx \theta$)	**long** ($\gg \theta$)
			
Angle ($\cos\angle(\mathbf{q},\mathbf{p})$)	small positive large positive	small positive large positive	small positive large positive
Suitable method	Prune	Cosine Similarity Search $\theta' = \frac{\theta}{\|\mathbf{q}\|\|\mathbf{p}\|}$	

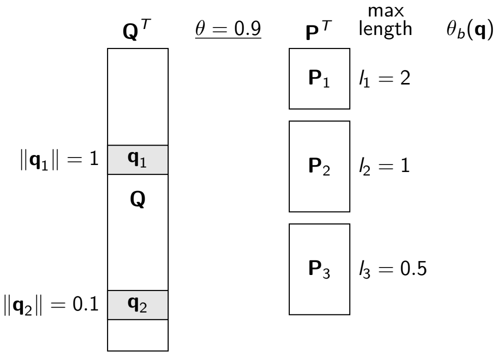
When is an inner product large?

$$\mathbf{q}^T\mathbf{p} = \|\mathbf{q}\|\|\mathbf{p}\|\cos\angle(\mathbf{q},\mathbf{p}) \geq \theta \ , \ -1 \leq \cos\angle(\mathbf{q},\mathbf{p}) \leq 1$$

Length ($\|\mathbf{q}\|\|\mathbf{p}\|$)	**short** ($< \theta$)	**medium** ($\approx \theta$)	**long** ($\gg \theta$)
			
Angle ($\cos\angle(\mathbf{q},\mathbf{p})$)	small positive large positive	small positive large positive	small positive large positive
Suitable method	Prune	Cosine Similarity Search $\theta' = \frac{\theta}{\|\mathbf{q}\|\|\mathbf{p}\|}$	Naive-like retrieval

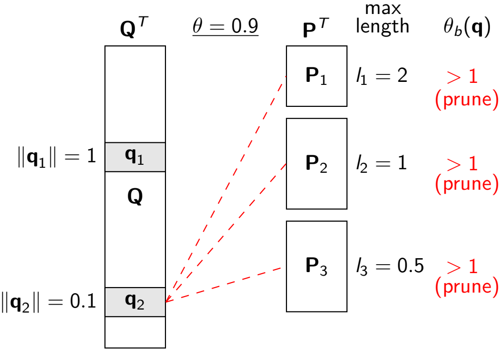
Main idea: bucketize by length

- Partition **P** in buckets with vectors of similar length
- Index buckets suitably



Main idea: bucketize by length

- Partition $\mathbf{P}$ in buckets with vectors of similar length
- Index buckets suitably
- For each query vector and bucket
  - Determine local threshold
  - Prune bucket if possible

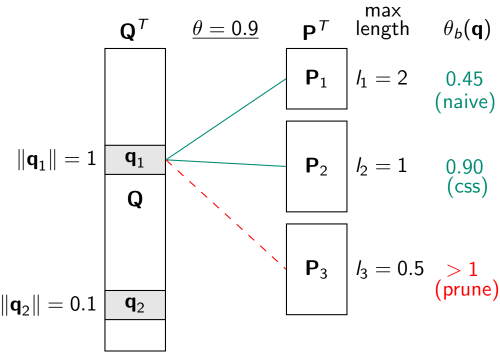


Local threshold on cosine similarity:

$$\theta_b(\mathbf{q}) = \frac{\theta}{\|\mathbf{q}\| l_b}$$

Main idea: bucketize by length

- Partition $\mathbf{P}$ in buckets with vectors of similar length
- Index buckets suitably
- For each query vector and bucket
  - Determine local threshold
  - Prune bucket if possible
  - Otherwise, select best retrieval method



Local threshold on cosine similarity:

$$\theta_b(\mathbf{q}) = \frac{\theta}{\|\mathbf{q}\| l_b}$$

Discussion

LEMP

- Prunes whole buckets
- Selects a suitable retrieval algorithm per query and bucket
- Can leverage existing methods
- Cache-friendly

Bucket-level retrieval

Choose among a variety of algorithms

- Fagin's Threshold Algorithm ($r < 10$)
- All pairs similarity search family ($1000 < r$)
- Space-partitioning trees ($r < 10$)

Our vectors are

- Not necessarily sparse
- Real values
- Medium dimensionality ($10 < r < 500$)

Two new algorithms

- COORD
- INCR

INCR: Main Idea

- $\mathbf{q}$, $\mathbf{p}$ qualify if $\|\mathbf{q}\|\|\mathbf{p}\|(\sum_{i=1}^{r} \bar{q}_i \bar{p}_i) \geq \theta$

$\|\mathbf{q}\|$	$\bar{\mathbf{q}}_1$	$\bar{\mathbf{q}}_2$	$\bar{\mathbf{q}}_3$	$\bar{\mathbf{q}}_4$	$\ldots$	$\bar{\mathbf{q}}_r$
$\|\mathbf{p}\|$	$\bar{\mathbf{p}}_1$	$\bar{\mathbf{p}}_2$	$\bar{\mathbf{p}}_3$	$\bar{\mathbf{p}}_4$	$\ldots$	$\bar{\mathbf{p}}_r$

INCR: Main Idea

- $\mathbf{q}$, $\mathbf{p}$ qualify if $\|\mathbf{q}\|\|\mathbf{p}\|(\sum_{i=1}^{r} \bar{q}_i \bar{p}_i) \geq \theta$
- Assume you have a budget of $\phi = 3$ multiplications for the $\sum_{i=1}^{r} \bar{q}_i \bar{p}_i$

$\|\mathbf{q}\|$	$\bar{\mathbf{q}}_1$	$\bar{\mathbf{q}}_2$	$\bar{\mathbf{q}}_3$	$\bar{\mathbf{q}}_4$	$\ldots$	$\bar{\mathbf{q}}_r$
$\|\mathbf{p}\|$	$\bar{\mathbf{p}}_1$	$\bar{\mathbf{p}}_2$	$\bar{\mathbf{p}}_3$	$\bar{\mathbf{p}}_4$	$\ldots$	$\bar{\mathbf{p}}_r$

INCR: Main Idea

- $\mathbf{q}$, $\mathbf{p}$ qualify if $\|\mathbf{q}\|\|\mathbf{p}\|(\sum_{i=1}^{r} \bar{q}_i \bar{p}_i) \geq \theta$

- Assume you have a budget of $\phi = 3$ multiplications for the $\sum_{i=1}^{r} \bar{q}_i \bar{p}_i$
- Goal: decide after seeing $\phi = 3$ coordinates:

$\|\mathbf{q}\|$	$\bar{\mathbf{q}}_1$	$\bar{\mathbf{q}}_2$	$\bar{\mathbf{q}}_3$	$\bar{\mathbf{q}}_4$	$\dots$	$\bar{\mathbf{q}}_r$
$\|\mathbf{p}\|$	$\bar{\mathbf{p}}_1$	$\bar{\mathbf{p}}_2$	$\bar{\mathbf{p}}_3$	$\bar{\mathbf{p}}_4$	$\dots$	$\bar{\mathbf{p}}_r$

INCR: Main Idea

- $\mathbf{q}$, $\mathbf{p}$ qualify if $\|\mathbf{q}\|\|\mathbf{p}\|(\sum_{i=1}^{r} \bar{q}_i \bar{p}_i) \geq \theta$

- Assume you have a budget of $\phi = 3$ multiplications for the $\sum_{i=1}^{r} \bar{q}_i \bar{p}_i$
- Goal: decide after seeing $\phi = 3$ coordinates: $\|\mathbf{q}\|\|\mathbf{p}\|(\sum_{i=1}^{3} \bar{q}_i \bar{p}_i + \mathit{UpperBoundFor}(\sum_{i=4}^{r} \bar{q}_i \bar{p}_i)) \geq \theta$

$\|\mathbf{q}\|$	$\bar{\mathbf{q}}_1$	$\bar{\mathbf{q}}_2$	$\bar{\mathbf{q}}_3$	$\bar{\mathbf{q}}_4$	$\dots$	$\bar{\mathbf{q}}_r$
$\|\mathbf{p}\|$	$\bar{\mathbf{p}}_1$	$\bar{\mathbf{p}}_2$	$\bar{\mathbf{p}}_3$	$\bar{\mathbf{p}}_4$	$\dots$	$\bar{\mathbf{p}}_r$

INCR: Main Idea

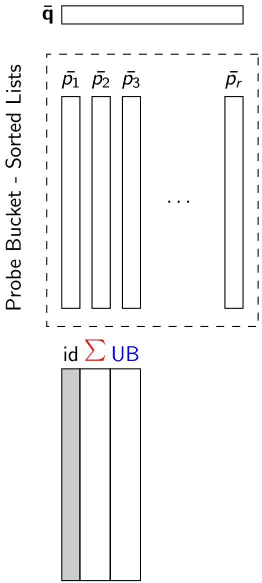
- $\mathbf{q}$, $\mathbf{p}$ qualify if $\|\mathbf{q}\|\|\mathbf{p}\|(\sum_{i=1}^{r} \bar{q}_i \bar{p}_i) \geq \theta$
- Assume you have a budget of $\phi = 3$ multiplications for the $\sum_{i=1}^{r} \bar{q}_i \bar{p}_i$
- Goal: decide after seeing $\phi = 3$ coordinates: $\|\mathbf{q}\|\|\mathbf{p}\|(\sum_{i=1}^{3} \bar{q}_i \bar{p}_i + \mathit{UpperBoundFor}(\sum_{i=4}^{r} \bar{q}_i \bar{p}_i)) \geq \theta$
- In practice
  - ϕ is automatically tuned
  - the ϕ coordinates do not have to be consecutive

$\|\mathbf{q}\|$	$\bar{\mathbf{q}}_1$	$\bar{\mathbf{q}}_2$	$\bar{\mathbf{q}}_3$	$\bar{\mathbf{q}}_4$	$\dots$	$\bar{\mathbf{q}}_r$
$\|\mathbf{p}\|$	$\bar{\mathbf{p}}_1$	$\bar{\mathbf{p}}_2$	$\bar{\mathbf{p}}_3$	$\bar{\mathbf{p}}_4$	$\dots$	$\bar{\mathbf{p}}_r$

INCR

$\bar{\mathbf{q}}$

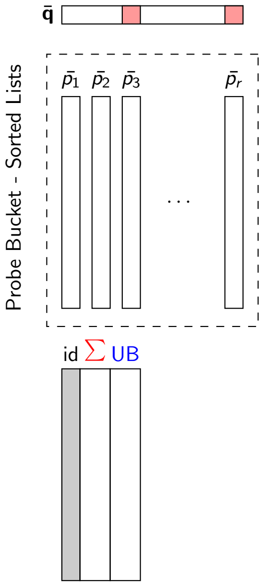
INCR



$\bar{\mathbf{q}}$
Probe Bucket - Sorted Lists
$\bar{p}_1$ $\bar{p}_2$ $\bar{p}_3$ $\bar{p}_r$
id $\sum$ UB

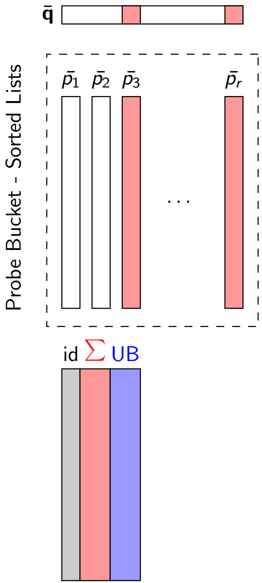
INCR

- Pick coordinates (largest $|\bar{q}_i|$ first)



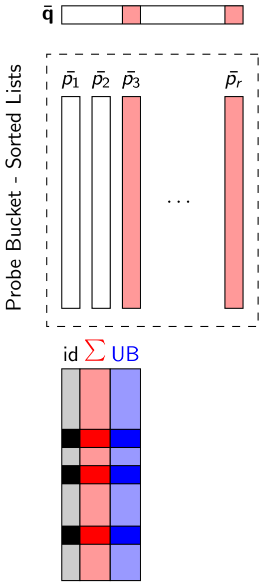
INCR

- Pick coordinates (largest $|\bar{q}_i|$ first)
- Scan indexes and update "$\sum$" and "UB" quantities



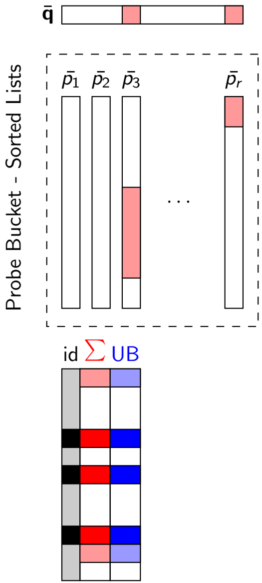
INCR

- Pick coordinates (largest $|\bar{q}_i|$ first)
- Scan indexes and update "$\sum$" and "UB" quantities
- Prune vectors for which $\|\mathbf{q}\|\|\mathbf{p}\|(\sum + UB) < \theta$



INCR

- Pick coordinates (largest $|\bar{q}_i|$ first)
- Scan indexes and update "$\sum$" and "UB" quantities
- Prune vectors for which $\|\mathbf{q}\|\|\mathbf{p}\|(\sum + UB) < \theta$
- No need to scan the whole lists! Bounds exist

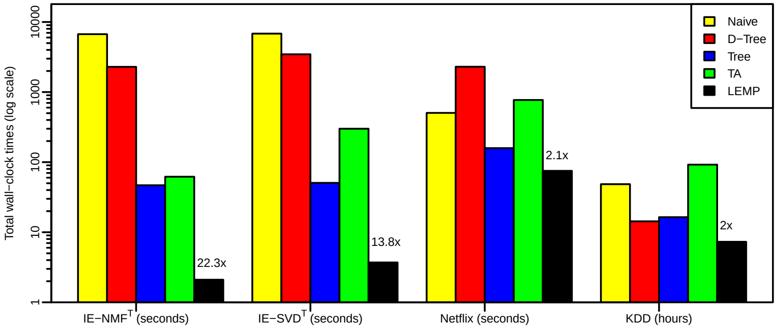


Discussion

INCR

- Computes partial inner products
- Uses simple index - cheap to construct
- Scans part of the index - does not necessarily scan from top
- Sequential memory access pattern, fast

How fast is it? (Top-1 movie per user)



# rows	132K	132K	480K	1000K
# cols	771K	771K	17K	624K
Sparse?	Yes	No	No	No
Length skew	Strong	Strong	Mild	Mild
Dimensions	50	50	50	50

Summary

Large inner-product search

- Matrix factorization common technique in data mining
- Large entries in matrix products are usually of particular interest

The LEMP algorithm

- Bucketizes vectors by length
- Prunes buckets whenever possible
- For the remaining buckets: selects efficient retrieval algorithms
- Consistently fastest bucket method: INCR

Summary

Large inner-product search

- Matrix factorization common technique in data mining
- Large entries in matrix products are usually of particular interest

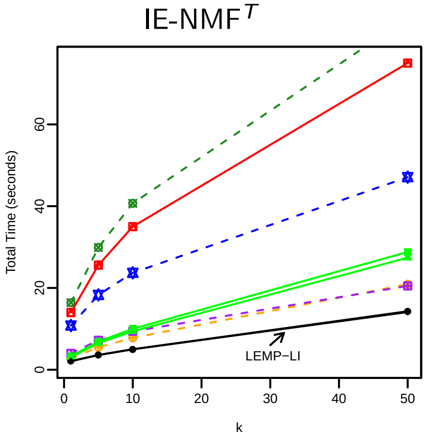
The LEMP algorithm

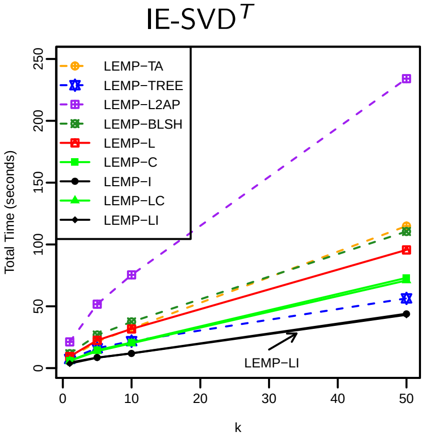
- Bucketizes vectors by length
- Prunes buckets whenever possible
- For the remaining buckets: selects efficient retrieval algorithms
- Consistently fastest bucket method: INCR

Thank you!

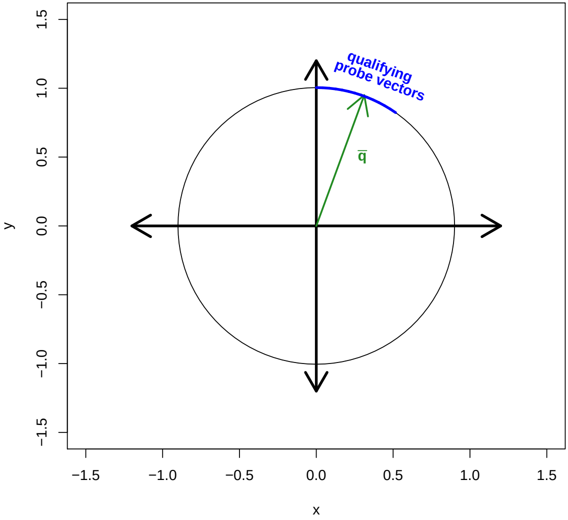
Questions?

Performance of bucket algorithms

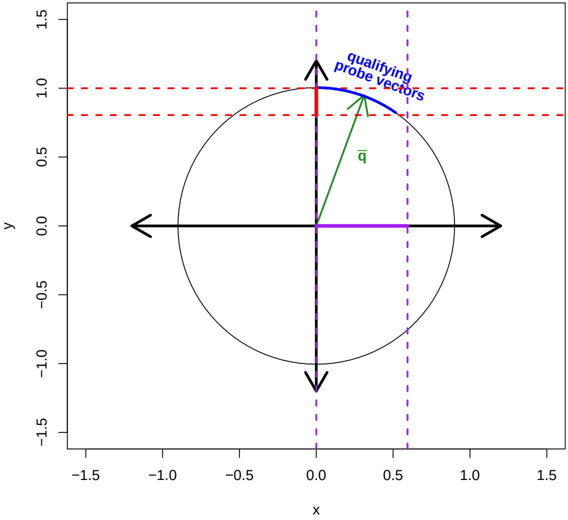




Do we need to scan the whole lists?

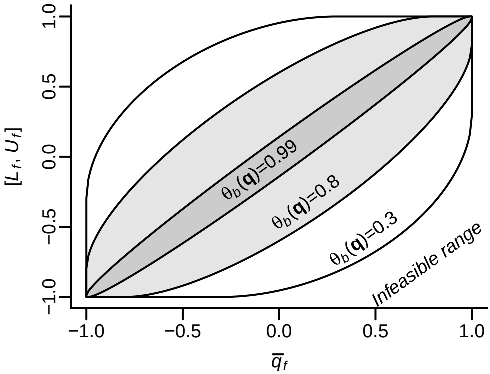


Do we need to scan the whole lists?



COORD: bounds

- Find for each coordinate $\bar{p}_f$ of $\bar{\mathbf{p}}$ an upper and lower bound $[L_f, U_f]$ such that $\bar{p}_f \notin [L_f, U_f] \Rightarrow \bar{\mathbf{q}}^T \bar{\mathbf{p}} < \theta_b(\mathbf{q})$



Algorithm selection

- Take a sample of queries
- Run both naive and INCR/COORD
- Estimate the value of t_b
- Linear classifier with $\theta_b(\mathbf{q})$ as feature