Pivot-Tabellen und Techniken zur Auswahl geeigneter Pivot-Elemente

(Effizienzsteigerung bei der Ähnlichkeitssuche)

Farzan Ranjbar Mirzakhani

Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Agenda

Agenda

- Motivation
- Lower Bound Distanzfunktion δ_{LB}
  - Wie wird sie eingesetzt?
  - Wie viel Kosten werden gespart?
  - Welche Eigenschaft muss sie erfüllen?
- Bestimmung von δ_{LB} und Einsatz von Pivots
- Kostenvergleich von δ_{LB} und δ
  - pro Objekt
  - pro Range-Query
- Kandidatenbestimmung mittels Lower Bound & Pivots (Beispiel)

Agenda

- Beeinflussung der Kandidatenliste durch Pivots
- Geeignete Auswahl von Pivot-Elementen
  - Hoher Pivot-Anzahl = Mehr Effizienz?
  - Effizienz-Kriterium: Wann ist ein Pivot-(Set) gut?
  - μD ermitteln
- Die unterschiedlichen Pivot-Auswahltechniken
  - Selection of N random groups
  - Incremental selection
- Auswirkung der Parameter A,N & k
- Konstruktionskosten des Pivot-Raums

Agenda

- Pivot-Auswahltechniken im Vergleich
- Schlussfolgerung
- Literatur- und Abbildungsverzeichnis

Motivation

Beispielszenario

- Eine Datenbank DB bestehend aus Bilderobjekten $o_i \in DB$

- Eine teure Distanzfunktion für den Vergleich zweier Bilder:
  $\delta(o_i , o_j)$

- Eine Bereichs-Anfrage (Range-Query):
  alle Bilder mit einem Ähnlichkeitsdistanz kleiner als r zu einem Bild q werden angefordert → RQ(q, r)

Beispielszenario

- Mithilfe von δ das Bild q mit allen Bildern $o_i \in$ DB vergleichen
  → |DB| x δ-Berechnungen nötig!
- Distanzfunktion δ, die exakte Distanzwerte liefert, ist jedoch teuer!
- Lösung: Lower Bound Distanzfunktion und Pivots

Lower Bound Distanzfunktion δ_{LB}

Wie wird sie eingesetzt?

- Mittels der günstigen δ_{LB} die Objekte aus der DB filtern, die als Antwort in Fragen kommen → Kandidatenliste erzeugen
- Anschließend auf diese Kandidaten die teure δ ausführen, um so falsche Antworten zu eliminieren
- → Übrig nur noch die geforderten Antworten

Wie viel Kosten werden gespart?

- Kosten pro Anfrage (mit Filterung):
  Kosten(Filterung) + |Kandidat| x δ-Berechnungen

- Kosten (ohne Filterung):
  |DB| x δ-Berechnungen

- → Je kleiner die Kandidatenliste bzw. |Kandidat| ist,
  desto geringer die Kosten im Vergleich zum Normalfall (ohne Filterung)

- → Je höher die Anzahl an Anfragen,
  desto bemerkbarer macht sich die eingesparte Zeit

Welche Eigenschaft muss sie erfüllen?

- Damit bei der Anfrage RQ(q, r) nur Objekte gefiltert werden, die $\delta(q, o_i) > r$ ergeben, muss $\delta_{LB}(q, o_i) \leq \delta(q, o_i)$ gelten

- Erfüllt ein Objekt $\delta_{LB}(q, o_i) > r$ → Erfüllt es ebenso $\delta(q, o_i) > r$

- Somit wird ein Objekt herausgeworfen, das auch $\delta(q, o_i) > r$ erfüllt und daher nicht $\delta(q, o_i) < r$ erfüllen wird
  → Keine Antwort-Objekte werden gefiltert!

Bestimmung von δ_{LB} und Einsatz von Pivots

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Bestimmung von δ_{LB}

- Distanzfunktion δ, die den Distanz zweier Objekte angibt, besitzt folgende drei Eigenschaften:
  - Strict positiveness: $\delta(q, o_i) = 0 \leftrightarrow q = o_i$
  - Symmetry: $\delta(q, o_i) = \delta(o_i, q)$
  - Triangle inequality: $\delta(q, o_i) \leq \delta(q, o_k) + \delta(y, o_i)$
- Durch die Dreiecksungleichung lässt sich folgende (nützliche) Ungleichung herleiten:
  - $\delta(q, o_i) \geq |\delta(o_i, o_k) - \delta(q, o_k)|$
- → Der Distanz $\delta(q, o_i)$ kann durch ein Lower Bound (zwei Distanzen $(\delta(o_i, o_k)$ & $\delta(q, o_k))$ abgeschätzt werden!

Rolle der Pivots

- An dieser Stelle kommen sogenannten Pivots zum Einsatz

- **Pivot** p_j ist ein ausgewähltes Objekt der DB, zu dem die Distanzen aller DB-Objekte (also $\delta(o_i, p_j)$) abgespeichert sind

- Man kann also
  $|\delta(o_i, p_j) - \delta(q, p_j)|$ als $\delta_{LB}(q, o_i)$ wählen

Rolle der Pivots

- Es gilt: Je näher δ_{LB} dem eigentlichen δ ist, desto weniger „falsche“ Antworten werden als Kandidaten ausgewählt

  Grund: Bei der Filterung mit $\delta_{LB}(q, o_i) > r$ werden bei kleineren δ_{LB} weniger Objekte rausgeschmissen

- → Daher wird bei k Pivots für den $\delta_{LB}(q, o_i)$
  - $\delta_{LB}(q, o_i) = \max_{1 \leq j \leq k} |\delta(o_i, p_j) - \delta(q, p_j)|$

    anstatt

  - $\delta_{LB}(q, o_i) = |\delta(o_i, p_j) - \delta(q, p_j)|$

Kostenvergleich von δ_{LB} & δ

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Kostenvergleich pro Objekt

- $\delta_{LB}(q, o_i)$ ist viel günstiger als $\delta(q, o_i)$
  Grund:
  - Lediglich die Distanzen des Query-Objekts zu Pivots berechnen (einmalig)

  → |Pivots| x δ-Berechnungen
  - Sonst für jedes $\delta_{LB}(q, o_i)$ nur die vorhandene Distanzen $\delta(q, p_j)$ & $\delta(o_i, p_j)$ voneinander subtrahieren

  → |Pivots| x Subtraktionen

  - Anstatt (ohne Filterung) für jedes Objekt

  → 1 x $\delta(q, o_i)$

Kostenvergleich pro Range-Query

- $\delta_{LB}(q, o_i)$ ist viel günstiger als $\delta(q, o_i)$
  Grund:
  - Lediglich die Distanzen des Query-Objekts zu Pivots berechnen (einmalig)
  → |Pivots| x δ-Berechnungen
  - Bei |DB| vielen Objekten
  → + |DB| x |Pivots| x Subtraktionen
  - Bei |Kandidat| große Kandidatenliste
  → + |Kandidat| x δ-Berechnungen

  - Anstatt (ohne Filterung) pro Anfrage
  → |DB| x δ-Berechnungen (bei großer |DB| sehr teuer!)

Kandidatenbestimmung mittels Lower Bound & Pivots (Beispiel)

Die folgenden Folien für das Beispiel
(Kandidatenbestimmung mittels Lower Bound & Pivots)
sind aus den DIS-Vorlesungsfolien von Dr. Fabian Panse entnommen

Kandidatenbestimmung

Given: Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1, 2, 2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta(\cdot, p_1)$	$p_2 = (1,3,0)$ $\delta(\cdot, p_2)$	$p_3 = (1,1,1)$ $\delta(\cdot, p_3)$	
o_1	3	1	4	
o_2	5	2	1	
o_3	4	4	2	
$\delta(q, \cdot)$				

Kandidatenbestimmung

Given: Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1, 2, 2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta(\cdot, p_1)$	$p_2 = (1,3,0)$ $\delta(\cdot, p_2)$	$p_3 = (1,1,1)$ $\delta(\cdot, p_3)$	
o_1	3	1	4	
o_2	5	2	1	
o_3	4	4	2	
$\delta(q, \cdot)$	3	3	2	

Compute distances between q and pivot objects

Kandidatenbestimmung

Given: Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1,2,2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta^{\triangle}_{p_1}(q,\cdot)$	$p_2 = (1,3,0)$ $\delta^{\triangle}_{p_2}(q,\cdot)$	$p_3 = (1,1,1)$ $\delta^{\triangle}_{p_3}(q,\cdot)$	
o_1	0	2	2	
o_2	2	1	1	
o_3	1	1	0	
$\delta(q,\cdot)$	3	3	2	

Compute $\delta^{\triangle}_{p_i}(q, o_i)$ for every object $o_i \in \mathbb{D}$ and pivot object $p_i \in \mathbb{P}$

➡ $|\delta(q,p_j) - \delta(o_i,p_j)|$

Kandidatenbestimmung

Given: Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1,2,2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta^{\triangle}_{p_1}(q,\cdot)$	$p_2 = (1,3,0)$ $\delta^{\triangle}_{p_2}(q,\cdot)$	$p_3 = (1,1,1)$ $\delta^{\triangle}_{p_3}(q,\cdot)$	$\delta^{\triangle}_{\mathbb{P}}(q,\cdot)$
o_1	0	2	2	2
o_2	2	1	1	2
o_3	1	1	0	1
$\delta(q,\cdot)$	3	3	2	

Compute $\delta^{\triangle}_{\mathbb{P}}(q, o_i) = \max_{p_i \in \mathbb{P}} \left(\delta^{\triangle}_{p_i}(q, o_i)\right)$ for every object $o_i \in \mathbb{D}$

Kandidatenbestimmung

Given: Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1, 2, 2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta^{\triangle}_{p_1}(q,\cdot)$	$p_2 = (1,3,0)$ $\delta^{\triangle}_{p_2}(q,\cdot)$	$p_3 = (1,1,1)$ $\delta^{\triangle}_{p_3}(q,\cdot)$	$\delta^{\triangle}_{\mathbb{P}}(q,\cdot)$
o_1	0	2	2	2
o_2	2	1	1	2
o_3	1	1	0	1
$\delta(q,\cdot)$	3	3	2	

Select every object o_i where $\delta^{\triangle}_{\mathbb{P}}(q, o_i) \leq \epsilon$ as candidate

Kandidatenbestimmung

→ Nun muss auf o_3 die teure $\delta(q, o_3)$ ausgeführt werden, um festzustellen, ob der Kandidat tatsächlich eine gewünschte Antwort auf die Anfrage darstellt

Beeinflussung der Kandidatenliste durch Pivots

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Beeinflussung der Kandidaten durch Pivots

- Steigende Pivot-Anzahl = kleinere Kandidatenliste

  Grund: Da aufgrund höhere Pivot-Anzahl mehr Distanzen $\delta(o_i, p_j)$ vorhanden sein werden, wird auch die Wahrscheinlichkeit für die folgende Filterungsungleichung steigen

  $$\max_{1 \leq j \leq k} |\delta(o_i, p_j) - \delta(q, p_j)| = \delta_{LB}(q, o_i) > r$$

Beeinflussung der Kandidaten durch Pivots

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta^{\Delta}_{p_1}(q,\cdot)$	$p_2 = (1,3,0)$ $\delta^{\Delta}_{p_2}(q,\cdot)$	$p_3 = (1,1,1)$ $\delta^{\Delta}_{p_3}(q,\cdot)$	$\delta^{\Delta}_{\mathbb{P}}(q,\cdot)$
o_1	0	2	2	2
o_2	2	1	1	2
o_3	1	1	0	1
$\delta(q,\cdot)$	3	3	2	

- Wäre der Pivot p_1 nicht da (ein Pivot weniger) → o_2 und o_3 Kandidaten, anstatt nur o_3
- (weniger Pivots = mehr Kandidaten)

Geeignete Auswahl von Pivot-Elementen

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Hoher Pivot-Anzahl = Mehr Effizienz?

Bei steigender Anzahl an Pivots...

– ...steigen die Anzahl an $\delta(q, p_j)$ Berechnungen (wenn auch diese einmalig sind)

– ... steigen die Anzahl an zu speichernden Distanzen zwischen den Objekten zu der großen Menge an Pivots

– Verkleinert sich die Kandidatenliste ab einer bestimmten Anzahl an Pivots langsamer

→ Höhere Pivot-Anzahl $\neq$ Höhere Effizienz bei der Suche

Effizienz-Kriterium

- Kann die gleiche Anzahl an Pivots (jedoch andere Objekte als Pivots) für eine kleinere Kandidatenliste sorgen?

- Wie kann zwischen Pivot-Sets mit gleicher Anzahl an Pivots der bessere Set bestimmt werden?

→ Effizienz-Kriterium

Effizienz-Kriterium

- Beispiel zur Erläuterung vom Effizienz-Kriterium
- DB-Objekte wurden in einen Pivot-Raum eingeordnet:

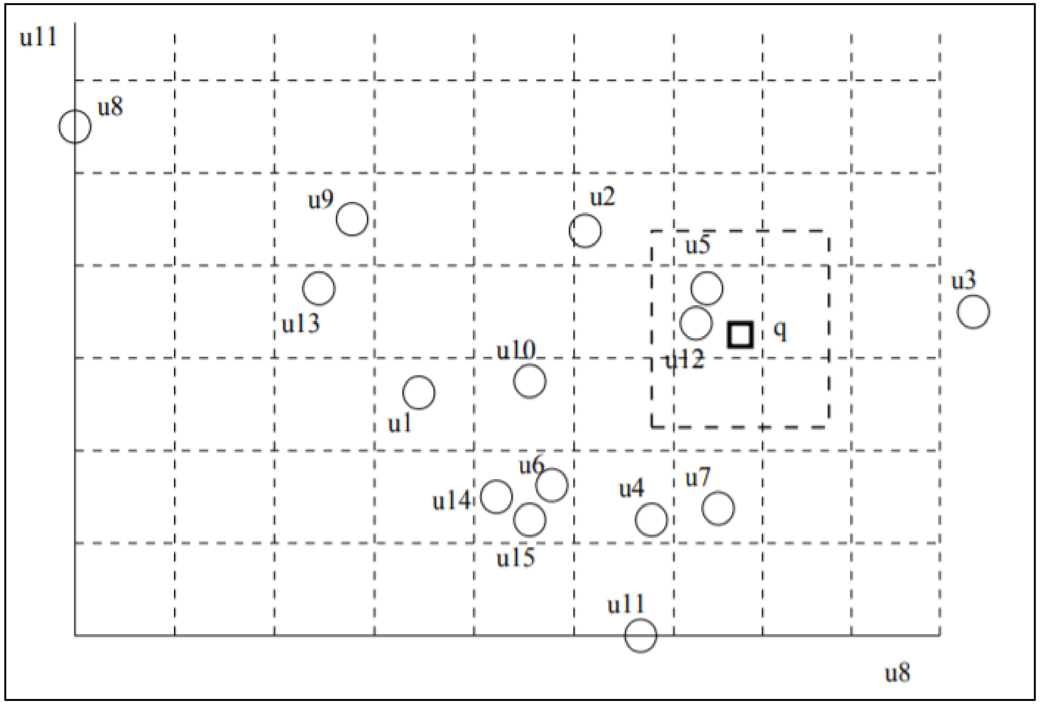

Abbildung 1: Pivot-Raum inklusive eingeordnete DB-Objekte

Effizienz-Kriterium

- Um die Kandidatenliste zu verkleinern, muss demnach die Wahrscheinlichkeit dafür, dass die Objekte im Quadrat-Raum liegen, verkleinert werden
- Dies ist dann der Fall, wenn der Max.-Abstand zweier Objekten im Durchschnitt groß ist (Objekte sind weit im P-Raum verstreut, somit ist die Wahrscheinlichkeit dafür, dass sich viele Objekte in einem Quadrat-Raum aufhalten kleiner)

$\rightarrow \max_{1 \leq j \leq k} |\delta(o_i, p_j) - \delta(o_k, p_j)|$ soll so groß wie möglich sein!

Effizienz-Kriterium

- Der durchschnittliche Maximal-Abstand der Objekte im P-Raum zueinander wird fortfolgend als µD bezeichnet

- Effizienz-Kriterium:

  Bei mehreren Pivot-Sets gleicher Kardinalität ist der Pivot-Set, bei dem die Objekte im jeweiligen P-Raum den höchsten µD besitzen, der beste Set unter den auszuwählenden Pivot-Sets!

μD ermitteln

- Um den besseren Pivot-Set wählen zu können, muss der μD ermittelt werden. Die Ermittlung von μD geschieht wie folgt:

1. Es werden zufällig A Paare $\{(a_1, a`_1), \ldots, (a_A, a`_A)\}$ aus der DB ausgewählt. Z.B. $\{(u_5, u_1), (u_{13}, u_7), \ldots\}$
2. Die Objekte, die in der Menge der Paare auftauchen, werden in den P-Raum abgebildet. (2A Objekten & k Pivots = 2Ak δ-Berechnungen)
3. Der Maximal-Abstand eines Paares $(a_i, a`_i)$ (als D_i bezeichnet), wird aus den vorhandenen Distanzen berechnen:

$$D_i = \max_{1 \leq j \leq k} |\delta(a_i, p_j) - \delta(a`_i, p_j)|$$

4. Die Summe der Max.-Abstände aller A Paare durch ihre Anzahl:

$$\frac{\sum_{i=1}^{A} D_i}{A}$$

Die unterschiedlichen Pivot-Auswahltechniken

Selection of N random groups

1. N zufällige Pivot-Sets werden erstellt (bestehend jeweils aus k Pivots)

2. Die µD zu den N Gruppen bzw. Pivot-Sets werden ermittelt
   (Bei N Gruppen mit k Pivots & A gewählten Paaren:
   <u>2AkN δ-Berechnungen</u> nötig, um die k gute Pivots auszuwählen)

3. Die Gruppe bzw. Pivot-Set mit dem höchsten uD wird als der Pivot-Set für den P-Raum gewählt

Incremental selection

Aus N zufällig zur auswahlstehenden Pivots wird das Pivot, das bei den A Paaren für den maximalen µD sorgt, als das Pivot p_1 ausgewählt

µD für das Pivot p_1 mit A Paaren wird wie folgt berechnet:

$$\frac{\sum_{i=1}^{A} |\delta(a_i, p_1) - \delta(a'_i, p_1)|}{A}$$

→ 2A δ-Berechnungen → Bei N Pivot-Proben = 2AN δ-Berechnungen

Incremental selection

Möchte man ein zweites Pivot hinzufügen (P-Raum mit zwei Pivots):

Aus N zufällig zur auswahlstehenden Pivots wird das Pivot als p_2 gewählt, das mit dem zuvor gefundenen p_1 für den höchsten µD sorgt

ACHTUNG: p_1 ist hierbei zuvor als gut ausgewählt und somit fixiert!

Incremental selection

Bei dem Hinzufügen des x-ten Pivot p_x
(beim zweiten Pivot wäre $p_x=p_2$)

...wird μD für den p_1, ..., p_x mit A Paaren wie folgt berechnet:

$$\frac{\sum_{i=1}^{A} \max \left|\, \max_{1 \leq j \leq x-1} \left|\delta(a_i, p_j) - \delta(a'_i, p_j)\right| \;,\; \left|\delta(a_i, p_x) - \delta(a'_i, p_x)\right| \,\right|}{A}$$

Incremental selection

Werden die Distanzen

$$\max_{1 \leq j \leq x-1} |\delta(a_i, p_j) - \delta(a'_i, p_j)| \qquad \text{(für alle Paare } i \in 1, \ldots, A)$$

die bei der Bestimmung von µD (beim Hinzufügen von Pivot p_{x-1} berechnet wurden) festgehalten. So müssen zur Bestimmung von µD beim Hinzufügen von p_x lediglich die folgenden Distanzen bestimmt werden:

$$|\delta(a_i, p_x) - \delta(a'_i, p_x)| \qquad \text{(für alle Paare } i \in 1, \ldots, A)$$

→ 2A δ-Berechnungen → Bei N unterschiedlichen Pivots, die als p_x abgecheckt werden: 2AN δ-Berechnungen

Incremental selection

2AN δ-Berechnungen, um ein gutes Pivot für den P-Raum auszuwählen
(Für einen P-Raum mit k guten Pivots → 2AkN δ-Berechnungen)

Erkennbarer Vorteil:

- Bei Erhöhung der Pivot-Anzahl werden die bis dahin gemachte Berechnungen nicht unbedeutend (wie bei anderen Techniken).
  → So werden bei der Neukonstruktion des P-Raums (mit mehr Pivots) enorm an Kosten gespart

Auswirkung der Parameter A,N & k

Die Parameter A,N und k

Je nachdem wie die folgenden Parameter gewählt werden, haben sie unterschiedliche Einflüsse, auf die Suche nach guten Pivots bzw. das Filtern von Objekten:

- **A** (Anzahl der zufällig ausgewählten Paare aus der DB):
  Je mehr Paare ausgewählt werden, desto präzisere µD-Bewertung
- **N** (Versuchsanzahl, bei der andere Pivots als gut überprüft werden):
  Je öfters unterschiedliche Pivots abgecheckt werden, desto bessere Pivots können gefunden werden
- **k** (Anzahl an gute Pivots, die gesucht werden):
  Je mehr Pivots im P-Raum sind, desto mehr Distanzen von DB-Objekten zu den Pivots sind vorhanden → bessere Filterung → kleinere Kandidatenliste

Konstruktionskosten des P-Raums

Konstruktionskosten des P-Raums

- Hat man die Pivots mittels einer Technik ausgewählt und die Distanzen der DB-Objekte zu ihnen berechnet und abgespeichert, hat man einen einmaligen Kosten von:

  (Pivot-Auswahl-Kosten) + (Kosten, um die restlichen DB-Objekte, die nicht als Paar gewählt wurden, auch in P-Raum einzuordnen)

- Nach den entstandenen Konstruktionskosten des P-Raums kann mit steigender Anzahl an Anfragen immer mehr Kosten gespart werden

Pivot-Auswahltechniken im Vergleich

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Auswahltechniken im Vergleich

- Wie viel δ-Berechnungen (Distanzevaluationen) pro Anfrage nach dem Filtern notwendig sind, um eine Anfrage zu beantworten, wird anhand einer Experiment-Umgebung und mithilfe einer Grafik fortfolgend klargestellt. So sollen auch die unterschiedlichen Pivot-Auswahltechniken miteinander verglichen werden.

  → Je geringer die Anzahl an δ-Berechnung, desto kleiner ist die Kandidatenliste, die wegen den gut ausgewählten Pivots gebildet worden ist

  → Je geringer die Anzahl an δ-Berechnung, desto besser ist demnach auch die jeweilige Auswahltechnik

Auswahltechniken im Vergleich

- Beim folgenden Experiment wurde ein DB mit 10.000 synthetische Objekte erstellt
- Die Objekte sind hierbei Punkte in einem 24-dimensionalen Vektorraum (etwa gleichverteilt)
- Die δ-Berechnung zwischen den Punkten erfolgt mittels Euklidischer Distanz-Funktion
- Die Range-Queries wurden hierbei so gestellt, dass bei einer Anfrage etwa 0.01% der DB-Objekte als Antwort erwartet wurden

Auswahltechniken im Vergleich

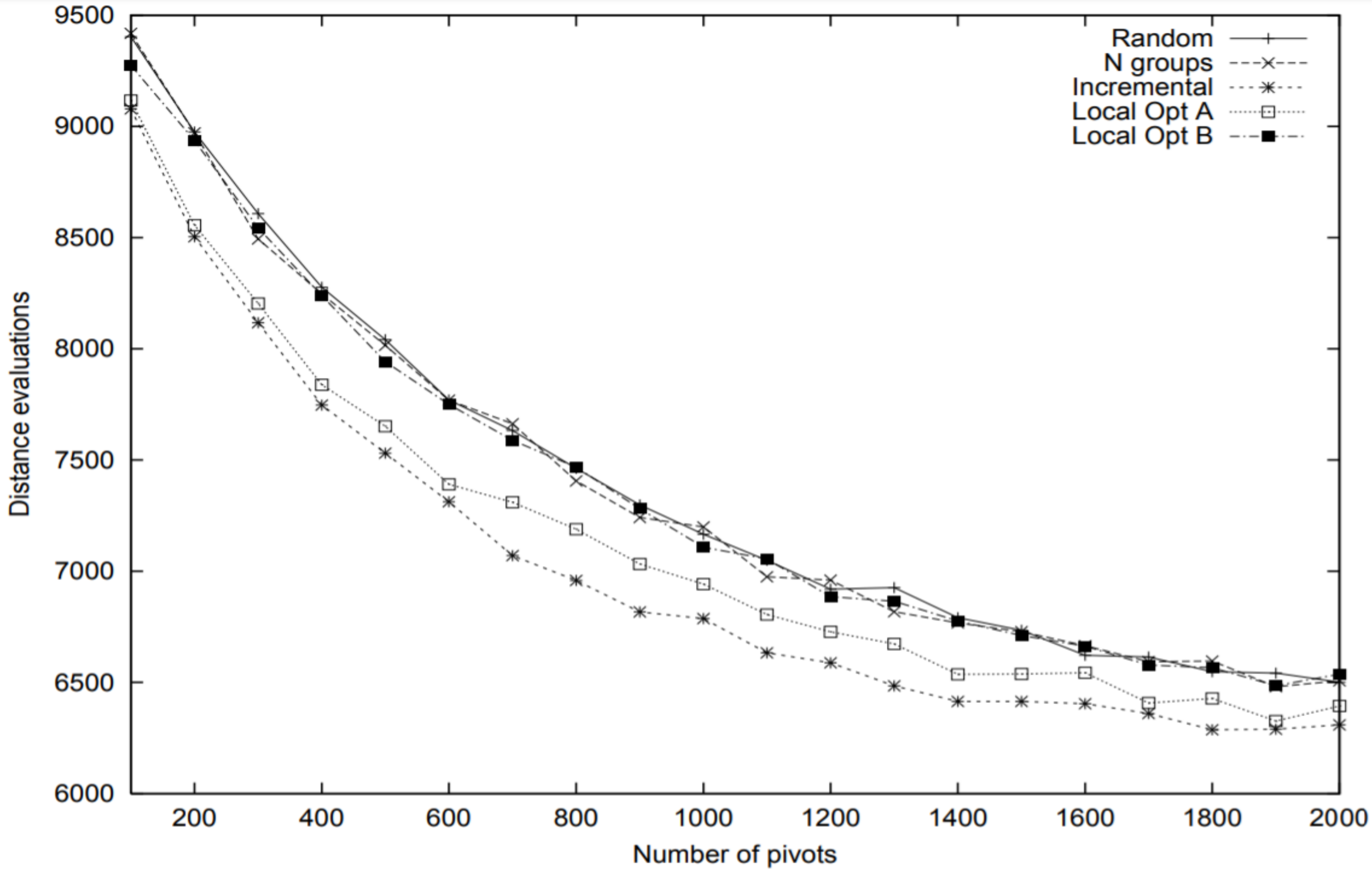

Abbildung 2: Vergleich der Auswahltechniken anhand einer Experiment-Umgebung

Auswahltechniken im Vergleich

+ Bei steigender Anzahl an k Pivots senkt selbstverständlich die Anzahl an Distanzevaluationen:

Da mehr Objekte gefiltert werden und somit die Kandidatenliste, auf die die δ-Berechnungen stattfinden, kleiner wird

- Ab einem k ist die Erhöhung jedoch nicht effizient:

Da Kandidatenliste nicht kleiner wird und mehr Distanzen abgespeichert werden müssen...

→ Wie aus der Abbildung 2 entnommen werden kann, ist die Auswahltechnik „Incremental“ die beste unter den verglichenen Techniken

Schlussfolgerung

Was kann man schlussfolgernd festhalten?

Schlussfolgerung

- Wir haben gesehen, dass zur effizienten Pivot-Auswahl ein Effizienz-Kriterium definiert werden kann: (μ_D maximal) → So können Auswahltechniken aufbauend auf diesem Kriterium gute Pivots für eine DB auswählen
- Unterschiedliche Techniken zur Auswahl von Pivots wurden vorgestellt
- Es wurde klargestellt, dass hohe k nicht unbedingt effizientere Suche bedeutet
- Zudem wurde festgestellt, dass bei der Ähnlichkeitssuche mittels Pivot-Tabellen zwar Konstruktionskosten anfallen, diese jedoch mit steigender Anzahl an Anfragen (nachhaltig) enorm an Kosten sparen

Literatur- und Abbildungsverzeichnis

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

VSIS

Literaturverzeichnis

- **[1]** Benjamin Bustos, Gonzalo Navarro, Edgar Chávez: Pivot Selection Techniques for Proximity Searching in Metric Spaces (2003), Abgerufen am 28.06.19 von: https://users.dcc.uchile.cl/~gnavarro/ps/sccc01.1.pdf

- **[2]** Lu Chen, Yunjun Gao, Baihua Zheng, Christian S. Jensen, Hanyu Yang, Keyu Yang: Pivot-based Metric Indexing (2017), Abgerufen am 28.06.19 von: https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=4741&context=sis_research

- **[3]** Fabian Panse: Similarity Search in Multimedia Data, Abgerufen am 28.06.19 von: https://vsis-www.informatik.uni-hamburg.de/oldServer/teaching/ss-17/dis/folien/15-SimilaritySearchMultiMedia.pdf

Abbildungsverzeichnis

Vielen Dank für Eure Aufmerksamkeit

Farzan Ranjbar Mirzakhani

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG