HY463 - Συστήματα Ανάκτησης Πληροφοριών
Information Retrieval (IR) Systems

Ευρετηρίαση και Αναζήτηση Πολυμέσων
Multimedia Indexing & Searching

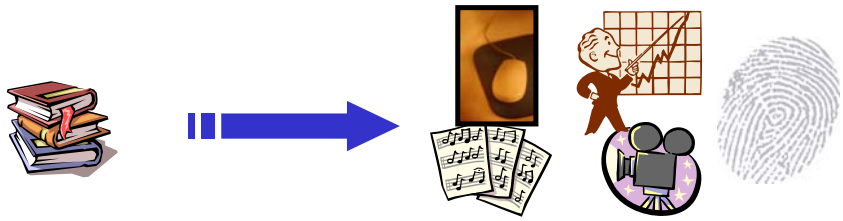
Γιάννης Τζίτζικας

Διάλεξη : 15
Ημερομηνία : 18-5-2007

Διάρθρωση Διάλεξης

- Το Πρόβλημα, Εφαρμογές και Παραδείγματα
- Επερωτήσεις Ομοιότητας
- Feature-based Indexing and Retrieval
  - Μέθοδοι Χωρικής Πρόσβασης
  - Case Study: 1D Time Series
  - Case Study: 2D Images
- Ανάκτηση Πολυμέσων χωρίς τη χρήση Features
  - μετρικά ευρετήρια

Το Πρόβλημα



Εδώ αντί για έγγραφα κειμένου έχουμε **πολυμέσα** (multimedia objects):

- **δισδιάστατες έγχρωμες εικόνες**
- **gray-scale ιατρικές εικόνες δισδιάστατες ή τρισδιάστατες (πχ MRI brain scans)**
- **one dimensional time series**
- **ψηφιοποιημένη φωνή ή μουσική**
- **video clips**

Στόχος: **Γρήγορη** εύρεση των πολυμέσων που ταιριάζουν ακριβώς ή προσεγγιστικά με μια επερώτηση.

Το απόλυτο ταίριασμα συνήθως δεν μας είναι χρήσιμο

Εφαρμογές και Παραδείγματα Επερωτήσεων

- Image Databases
  - q = ***Βρες εικόνες που μοιάζουν με ηλιοβασίλεμα***
    - (των οποίων η κατανομή χρώματος είναι ίδια με αυτήν των φωτογραφιών με ηλιοβασίλεμα)
  - αναγνώριση προσώπου (face recognition), ταίριασμα δακτυλικών αποτυπωμάτων (fingerprint matching)
- Financial, Marketing and Production Time Series
  - q = ***Βρες εταιρείες με παρόμοια διακύμανση μετοχών***
- Medical Applications
  - q = ***Βρες ιατρικές ακτινογραφίες που απεικονίζουν κάτι με υφή όγκου (tumor)***
- Audio/Video databases
  - αναγνώριση φωνής (voice recognition)
- DNA/Genome databases

Επερωτήσεις Ομοιότητας (Similarity Queries)

- Βασίζονται σε μια *δοθείσα* **συνάρτηση απόστασης** (ομοιότητας)

- Υπόβαθρο:
  - **Μετρικός χώρος** είναι ένα σύνολο στοιχείων **X** εφοδιασμένο με μία **συνάρτηση απόστασης** d: X x X -> R
  - Οι συναρτήσεις απόστασης έχουν τις εξής ιδιότητες, για οποιοδήποτε x,y,z στο X:
    - $d(x,y) \geq 0$ positiveness,
    - $d(x,y) = d(y,x)$ symmetry,
    - $d(x,x)=0$ reflexivity, and
    - $d(x,y) \leq d(x,z) + d(z,y)$ triangle inequality.
- Οι διανυσματικοί χώροι είναι μια ειδική περίπτωση των μετρικών χώρων

Παραδείγματα Συναρτήσεων Απόστασης για διανυσματικούς χώρους

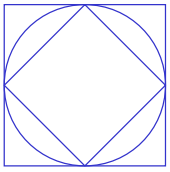
- There are a number of options for the distance function to use, but the most widely used is the family of L_s distances defined as:

$$L_s((x_1,...,x_k)(y_1,...,y_k)) = \left(\sum_{i=1}^{k} |x_i - y_i|^s \right)^{1/s}$$

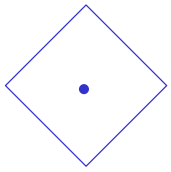
- L_1 :sum of the differences along the coordinates (also called "block" or "Manhattan" distance, since in two dimensions it corresponds to the distance to walk between two points in a city of rectangular blocks).
- L_2: "Euclidean" distance, as it corresponds to our notion of spatial distance.
- L_∞, : corresponds to taking the limit of the L_s formula when s goes to infinity.

Ισομετρικές καμπύλες $L_1, L_2 L_\infty$ για το δισδιάστατο χώρο

$$\sqrt[p]{(x^p + y^p)}$$

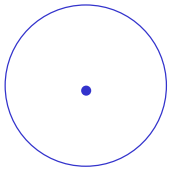


L_1



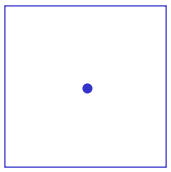
$$x + y = 1$$

L_2



$$\sqrt[2]{(x^2 + y^2)} = 1$$

L_∞



$$\lim_{p\to\infty} \sqrt[p]{(x^p + y^p)} = 1$$

$$\max(x, y) = 1$$

Κατηγορίες Επερωτήσεων Ομοιότητας

- **Ταιριάσματος** Προτύπου (Whole match queries)
  - Π.χ.: ***βρες όλα τα αντικείμενα σε απόσταση ≤ ε από το Q***
  - Τα αντικείμενα και η επερώτηση είναι ιδίου τύπου
    - π.χ. έγχρωμες εικόνες 512x512
- **Ταιριάσματος Υποπροτύπου** (Sub-pattern match queries)
  - Π.χ.: ***βρες όλα τα τμήματα των αντικειμένων σε απόσταση ≤ ε από το Q***
  - Η επερώτηση μπορεί να έχει διαφορετικό τύπο από τα αντικείμενα
    - π.χ. αντικείμενα = εικόνες 512x512, επερώτηση = τυπική 16x16 ακτινογραφία ενός όγκου
- **Κοντινότερων γειτόνων** (nearest neighbors)
  - Π.χ.: ***βρες τα 5 κοντινότερα αντικείμενα στο Q***
- **Σύνδεσης** (all pairs, spatial joins, ...)

Ταίριασμα Προτύπου: Ορισμός Προβλήματος

- **Σύμπαν** αντικειμένων U (πχ το σύνολο όλων των εικόνων 512x512)
- **Συλλογή** αντικειμένων $C=\{o_1,\ldots,o_N\}$, $C \subseteq U$
- Συνάρτηση **Απόστασης** (ομοιότητας,..,συνάφειας) **D:** $U \times U \rightarrow [0,1]$
  - καθορίζεται από έναν ειδικό του πεδίου (μπορεί να υπολογίζεται από ένα πρόγραμμα)
- **Επερώτηση**: *αντικείμενο* Q ($Q \in U$) και *ανοχή* (tolerance) ε
- **Απάντηση** επερώτησης: $ans(Q, \varepsilon)=\{ o \in C \mid D(o,Q) \leq \varepsilon \}$
- **Σκοπός**: *Γρήγορος* υπολογισμός του $ans(Q, \varepsilon)$

Μια Απλοϊκή Μέθοδος Αναζήτησης

1. Υπολόγισε το $D(o,Q)$ για κάθε o στο C
2. *Επέλεξε εκείνα τα αντικείμενα τ.ω. $D(o,Q) \leq \varepsilon$*

Παρατηρήσεις

- Πολύ αργή μέθοδος αν:
  - ο υπολογισμός της απόστασης ακριβός
    - For example, the **edit distance** in DNA strings requires a dynamic programming algorithm, which grows like the product of the string lengths (typically in the hundreds or thousands, for DNA databases)
  - το $|C|$ είναι μεγάλο

Ζητούμενα

- Ταχύτητα
  - η σειριακή σάρωση και ο υπολογισμός της απόστασης *d(o,Q)* για κάθε αντικείμενο *o* του *C* θα ήταν πολύ αργός (για μεγάλες βάσεις)
- Ορθότητα
  - υψηλός βαθμός ανάκλησης (να μην χάνεται κανένα από τα αντικείμενα),
  - καλός βαθμός ακρίβειας (τα εσφαλμένως ανακτηθέντα μπορούν να φιλτραριστούν με ένα επιπλέον πέρασμα)
- Μικρή χωρική επιβάρυνση
- Κλιμάκωση, Δυναμική:
  - εύκολη εισαγωγή, διαγραφή και ενημέρωση αντικειμένων

Διαφορές με το κλασσικό IR κειμένων

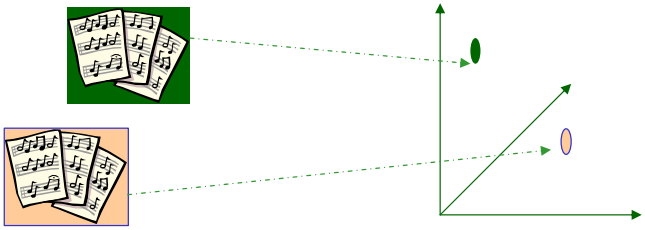
- Ο Βαθμός Συνάφειας εδώ αντιστοιχεί στο Βαθμό Ομοιότητας (Εγγύτητας) (1-Απόσταση)
  - Rel(o,Q) ≈ 1 - D(o,Q)
  - D(o,Q) ≈ 1 - Rel(o,Q)
- Η D(Q,o) εδώ **δίδεται**, στο text IR "εφευρίσκεται / ακαλύπτεται»
  - στα μοντέλα ανάκτησης κειμένων (εκτός του Boolean Μοντέλου) λαμβάνεται υπόψη ολόκληρη η συλλογή (θυμηθείτε τα βάρη idf, df). Με άλλα λόγια τα μοντέλα αυτά δεν ορίζουν το *D(d,Q)* αλλά το *D(**C**,d,Q)*
- Εδώ δίνουμε ιδιαίτερη έμφαση στην ταχύτητα αναζήτησης
- Subpattern match queries για text IR:
  - ***Βρες εκείνες τις παραγράφους των κειμένων που είναι συναφείς με την επερώτηση***

Ανάκτηση Πολυμέσων: Feature-based Approach

Feature:

- Ένας αριθμός που **χαρακτηρίζει** (υπό μια σκοπιά) ένα αντικείμενο, π.χ.
  - Μέσος Όρος,
  - Τυπική Απόκλιση (standard deviation),
  - Discrete Fourier transform (DFT)

- Χρησιμοποιώντας κ-features μπορούμε να δούμε τα αντικείμενα ως σημεία ενός κ-διάστατου χώρου



- Αναζήτηση = Αναζήτηση σε Πολυδιάστατους Χώρους

GEMINI: A Generic Multimedia Indexing Approach

- Βασική Ιδέα:
  - 1. ένα **quick-and-dirty** πέρασμα για απόρριψη των άσχετων αντικειμένων
    - βασίζεται στην έννοια του feature(s)
  - 2. χρήση μεθόδων **χωρικής πρόσβασης** για γρήγορη αναζήτηση

- Παράδειγμα
  - Έστω $C=\{s_1,\ldots, s_N\}$ όπου κάθε s_i είναι ένας πίνακας [1…365] ακεραίων
  - Ευκλείδεια συνάρτηση απόστασης:

$$D(s,Q)=\sqrt{\left(\sum_{i=1}^{365}(s[i]-Q[i])^2\right)}$$

  - Ο υπολογισμός του D(s,Q) απαιτεί 365 αφαιρέσεις, 365 πολλαπλασιασμούς
  - Ιδέα:
    - **χαρακτηρισμός κάθε σειράς με ΈΝΑ νούμερο**
      - για παράδειγμα μπορούμε να χαρακτηρίσουμε κάθε σειρά με το μέσο όρο των τιμών της
    - **με το μέσο όρο μπορούμε να απορρίψουμε πολύ γρήγορα τις σειρές που σίγουρα έχουν μεγάλη απόσταση με την επερώτηση**

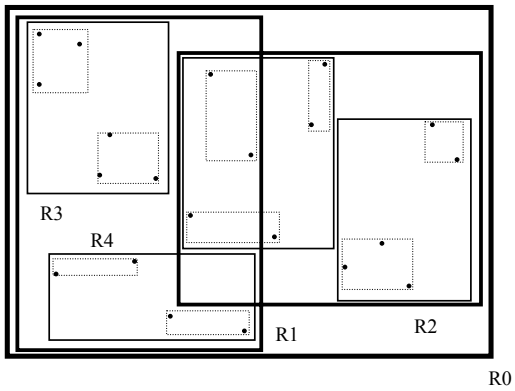
Επιλογή των Features

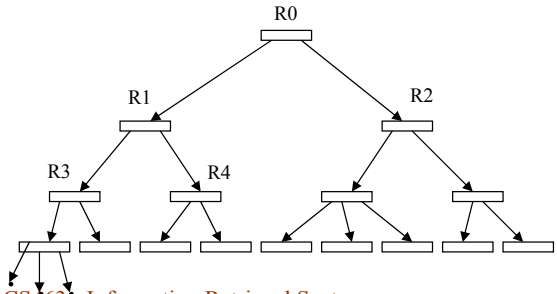
- Παρατηρήσεις
  - Ένα ***καλό*** Feature μας επιτρέπει με μια αριθμητική πράξη σύγκρισης να απορρίψουμε πολλά αντικείμενα
  - Μεγάλη διαφορά των Features **συνεπάγεται** μεγάλη διαφορά Σειρών
  - Μεγάλη διαφορά Σειρών **δεν συνεπάγεται πάντα** μεγάλη διαφορά Μ.Ο. (μέσου όρου), για αυτό μπορεί να έχουμε false positives (εσφαλμένως θετικά)
  - Με χρήση **πολλών features** μπορούμε να τα πάμε καλύτερα (λιγότερα false positives)
    - με χρήση *f* features κάθε αντικείμενο **o** αντιστοιχίζεται σε ένα σημείο *F(**o**)* του f-διάστατου χώρου,
  - Οργανώνοντας τα f-D σημεία με μια δομή χωρικής πρόσβασης μπορούμε να απορρίψουμε γρήγορα πολλά αντικείμενα (άρα δεν χρειαζόμαστε ούτε το quick-and-dirty στάδιο)

Μέθοδοι Χωρικής Πρόσβασης (Spatial Access Methods)

- SAM:
  - Δομές αναζήτησης για διανυσματικούς χώρους
  - they make extensive use of coordinate information to group and classify points in the space. For example, kd-trees divide the space along different coordinates and R-trees group points in hyper-rectangles
- Κύριες μέθοδοι
  - R-tree (R*-tree, X-trees, SR-trees, …)
  - Linear quadtrees

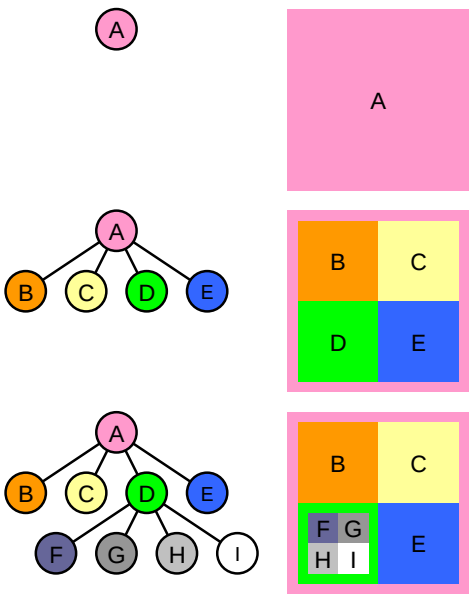
R-Tree





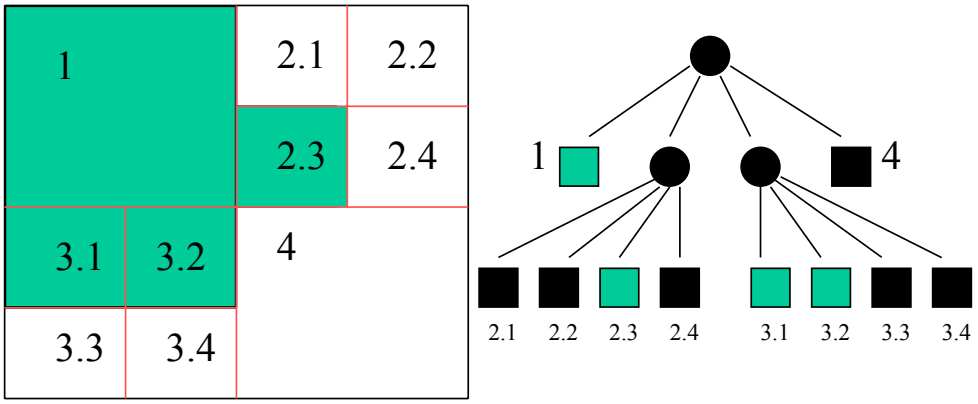
- Extension of B-tree to multidimensional space.
- Can support both point data and data with spatial extent (e.g., rectangles)
- Group objects into possibly overlapping clusters (rectangles in our case)
  - minimum bounding rectangle MBR
- Search of a range query proceeds along all paths that overlap with the query.
  - we recursively descend the R-tree, excluding branches whose MBRs do not intersect the query MBR

Quadtrees



- A *quadtree* is a tree in which each node has at most 4 children.
- Each node of the tree corresponds to a square (quadtree) region.
- If a node has children, think of its region being chopped into 4 (quadtree) subregions. Child nodes correspond to these smaller subregions of their parent's region.
- Subdivide as little or as much as is necessary.
- Each internal node has exactly 4 (quadtree) children.

Quadtrees (II)



Priority R-Trees (SIGMOD'04)

- See http://www.cs.duke.edu/~yike/prtree/pr.pdf

Οι αδυναμίες των SAMs

- They are very sensitive to the vector space dimensions.
- Closest point and range search algorithms have an exponential dependency on the dimension of the space (this is called the "**curse of dimensionality**").
- Vector spaces may suffer from large differences between their **representational dimension** (*k*) and their **intrinsic dimension**, i.e. the real number of dimensions in which the points can be embedded while keeping the distance among them.
  - For example, a plane embedded in a 50-dimensional space has intrinsic dimension 2 and representational dimension 50.
  - This is, in general, the case of real applications, where the data are clustered, and it has led to attempts to measure the intrinsic dimension such as the concept of "fractal dimension". Despite the fact that no technique can cope with intrinsic dimension higher than 20, much higher representational dimensions can be handled by dimensionality reduction techniques.

Αλγόριθμος Αναζήτησης

Προστάδιο:
Οργάνωσε τα f-D σημεία των αντικειμένων του C σε μια δομή χωρικής πρόσβασης

Είσοδος: Συλλογή C, Επερώτηση *Q,ε*
Έξοδος: *ans(Q,ε)*
1) Αντιστοίχισε το Q στο σημείο *F(Q)*

2) Χρησιμοποιώντας τη δομή χωρικής πρόσβασης βρες όλα τα σημεία σε απόσταση ε από το F(Q) στο F-χώρο:
TMP:= { o in C | ***d(F(o), F(Q))<ε*** } // Fast

3) Για τα ευρεθέντα υπολόγισε την πραγματική απόσταση από το Q και απέρριψε τα υπόλοιπα.
ΑΠΑΝΤΗΣΗ := { o **in TMP** | ***d(o, Q)<ε*** }

Εξασφάλιση ορθότητας
(αποτροπή λανθασμένων απορρίψεων)

Πότε ένα feature είναι καλό ?

- Ιδανική περίπτωση: Διατήρηση αποστάσεων:
  - ***D(o,o') = D(F(o), F(o'))***
- Αποδεκτή περίπτωση: Αποτροπή false-negatives (false-dismissals)
  - Ναι, αν η απεικόνιση *F()* φέρνει τα αντικείμενα πιο .. κοντά απ' ότι είναι στην πραγματικότητα, δηλαδή
  - ***D(F(o), F(o')) ≤ D(o,o') για όλα τα o,o' στο U.***
  - αν ισχύει αυτή η συνθήκη τότε δεν θα χάσουμε κανένα αντικείμενο

Case Study: One-Dimensional Time Series

- C = σύνολο σειρών ιδίου μήκους
- Πχ
  - *βρες τις εταιρίες με τιμές μετοχών όμοιες με αυτές της IBM*
- Συναρτήσεις απόστασης
  - Ευκλείδεια
- Παραδείγματα Features τα οποία μπορούμε να χρησιμοποιήσουμε
  - Μέσος όρος (M.O.)
  - M.O. 1ου εξαμήνου και M.O. 2ου εξαμήνου (άρα 2 features)
  - Coefficients of the Discrete Fourier Transform
  - for $x=(x_0,\ldots x_{n-1})$ let X_F denote the n-point DFT coefficient at the F-th frequency F=0,…,n-1. Άρα έχω $(X_0,\ldots,X_n)$

One-Dimensional Time Series (II)

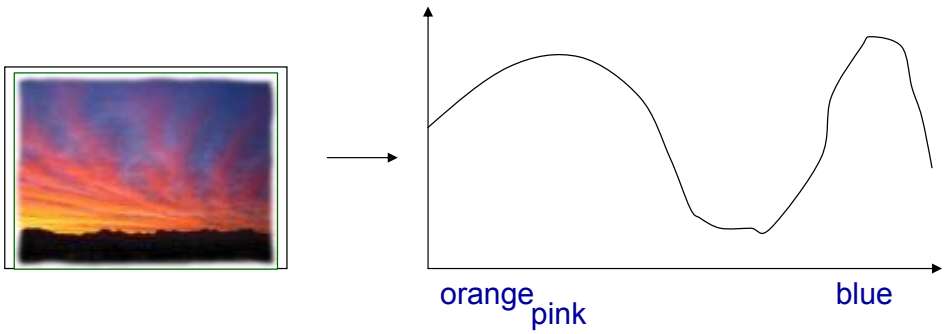
- Coefficients of the Discrete Fourier Transform
  - for $x=(x_0,\ldots x_{n-1})$ let X_Z denote the n-point DFT coefficient at the Z-th frequency Z=0,…,n-1.
  - Άρα για κάθε $x=(x_0,\ldots x_{n-1})$ έχω το $X=(X_0,\ldots,X_n)$
- Parvesal's theorem
  - D(x,y) = D(X,Y)
  - Άρα αυτό το feature είναι καλό (αφού διατηρεί τις αποστάσεις)
- Μπορούμε να χρησιμοποιήσουμε λιγότερες, π.χ. κ < n συχνότητες

$$D(F(x),F(y)) = \sum_{Z=0}^{k-1} |X_Z - Y_Z|^2 \leq \sum_{Z=0}^{n-1} |X_Z - Y_Z|^2 = \sum_{Z=0}^{n-1} |x_i - y_i|^2 = D(x,y)$$

έτσι δεν θα χάσουμε κανένα αντικείμενο (αφού με κ συχνότητες οι αποστάσεις μεταξύ των αντικειμένων είναι μικρότερες),
και συνάμα κρατάμε χαμηλά το πλήθος των διαστάσεων

Case Study: Δισδιάστατες έγχρωμες εικόνες

- Επερωτήσεις βάσει περιεχομένου
- Διαστάσεις περιεχόμενου
  - χρώμα, υφή, μορφές, θέσεις, ...
- Χρώμα
  - Για κάθε εικόνα υπολογίζουμε το k-element color histogram (k=256)



Δισδιάστατες έγχρωμες εικόνες (II)

- Απόσταση Ιστογραμμάτων
  - based on distance of colors matrix , etc
  - ο υπολογισμός της απόστασης είναι ακριβός
- Καλά Features:
  - first few coefficients of the two-dimentional DFT transform
  - RedGreenBlue: average amount of red, green blue
    - κάθε εικόνα απεικονίζεται σε ένα διάνυσμα (Ravg, Gavg, Bavg)
    - D(F(o),F(Q))= ευκλείδεια απόσταση των διανυσμάτων (Ravg, Gavg, Bavg)

Evaluating Sub-pattern match Queries

Evaluating Sub-pattern match Queries

- Π.χ.: ***βρες όλα τα <u>τμήματα</u> των αντικειμένων σε απόσταση ≤ ε από το Q***
- Assume time sequences
- Let *m* the length of the query pattern
- We slide a window of length *m* and for each position of the window we compute the *f* features
- Every sequence becomes a trail in the f-dimensional space
- This trail can be approximated by a MBR (minimum bounding rectangle)
- Representing each sequence by a few MBRs in feature space may allow false alarms but no false dismissals

Διάρθρωση Διάλεξης

- Το Πρόβλημα, Εφαρμογές και Παραδείγματα
- Επερωτήσεις Ομοιότητας
- Feature-based Indexing and Retrieval
  - Μέθοδοι Χωρικής Πρόσβασης
  - Case Study: 1D Time Series
  - Case Study: 2D Images
- **Ανάκτηση Πολυμέσων χωρίς τη χρήση Features**
  - μετρικά ευρετήρια

Ανάκτηση Πολυμέσων χωρίς τη χρήση Features

- Με χρήση **μετρικών ευρετηρίων** (metric indices, metric trees)
- Πρόκειται για τεχνικές που εφαρμόζονται κατευθείαν στις αποστάσεις
  - άρα δεν χρειαζόμαστε features
  - φτιάχνουν ιεραρχίες ομάδων, δενδρική δομή σφαιρών που περιλαμβάνουν άλλες σφαίρες, κ.ο.κ
- Key-point
  - Υπολογίζουμε τις αποστάσεις μεταξύ των αντικειμένων μια φορά, φτιάχνουμε μια κατάλληλη δομή δεδομένων, και εν συνεχεία την αξιοποιούμε κατά την αποτίμηση των επερωτήσεων (για να μειώσουμε το πλήθος των αποστάσεων που απαιτείται να υπολογίσουμε)

Κατηγορίες Μετρικών Ευρετηρίων

(A) tree indexes for *discrete* distance functions,
- i.e. for functions that deliver a small set of values
- **Burkhard-Keller Tree** (BKT) [Buthard et al 73]
- **Fixed Query Tree** (FQT) [Baeza-Yates 94]

(B) tree indexes for *continuous* distance functions
- i.e. for functions where the set of alternatives is infinite or very large
- **Vantage Point-Trees** (VTPs)
- **Multi-Vantage-Point trees** (MVTs)
- **Voronoi Trees** (VTs)
- **M-trees** (MT).

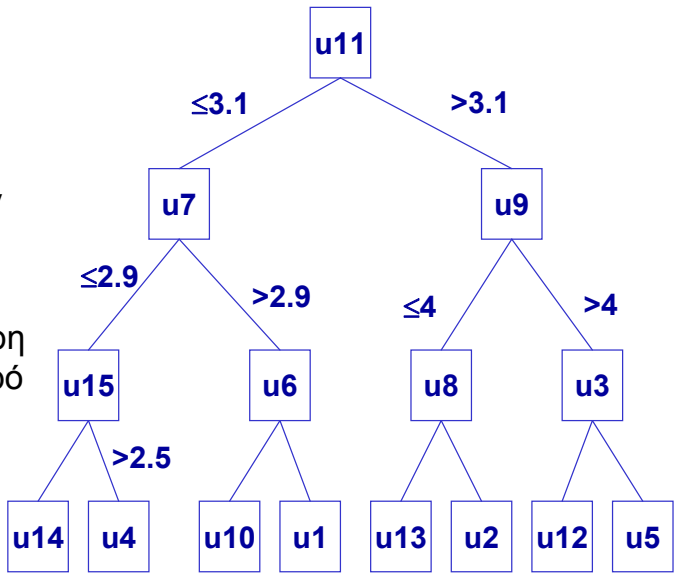
(C) not tree-based indexes.
- **AESA** (Approximating Eliminating Search Algorithm)
- **LAESA** (for linear AESA).

(B) Tree Indexes for Continuous Distance Functions

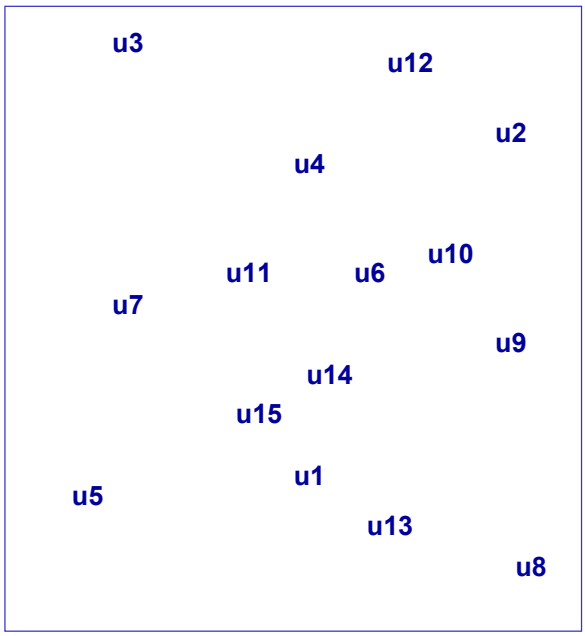
- If we have a continuous distance or if the distance function gives too many different values, it is not possible to have a child of the root for any such value.
- However the indexes for discrete functions can be adapted to a continuous distance by assigning a range of distances to each branch of the tree.
- Other examples of indexes for continuous distance functions include
  - Vantage Point-Trees (VTPs),
  - Multi-Vantage-Point trees (MVTs),
  - Voronoi Trees (VTs),
  - M-trees (MT).

Vantage-Point-Trees (VTPs) (also called metric-trees)

- Είναι δυαδικά δένδρα
- Τρόπος κατασκευής
  - Επιλέγουμε ένα στοιχείο κεντρικό (pivot).
  - Υπολογίζουμε τον μέσο όρο M των αποστάσεων από αυτό το σημείο
  - Τα στοιχεία με απόσταση μικρότερη ή ίση του M εισάγονται στο αριστερό υποδένδρο, ενώ τα υπόλοιπα στο δεξί
  - Συνεχίζουμε αναδρομικά

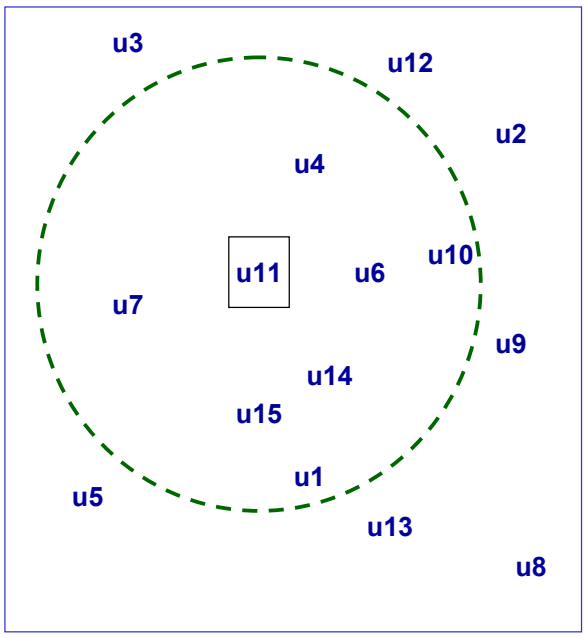


Παράδειγμα κατασκευής ενός VPT



u11,u7,u15,u6,u14,u4,u10,u1, u3,u12,u2,u9,u13,u8,u5

Επιλογή κεντρικού στοιχείου (pivot)

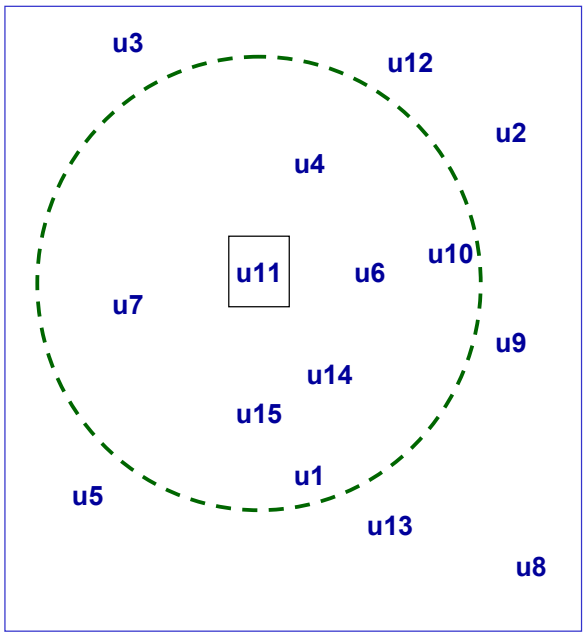


p=u11

median(d(p,u) | u in U}**=3.1**

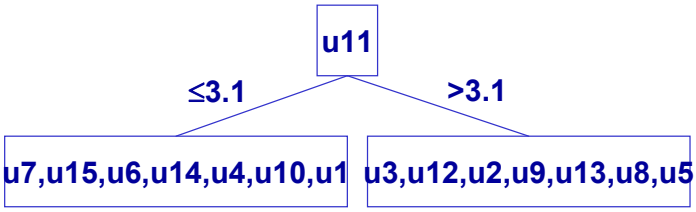
u11,u7,u15,u6,u14,u4,u10,u1, u3,u12,u2,u9,u13,u8,u5

Διαμέριση στοιχείων βάσει του Μ.Ο. των αποστάσεων

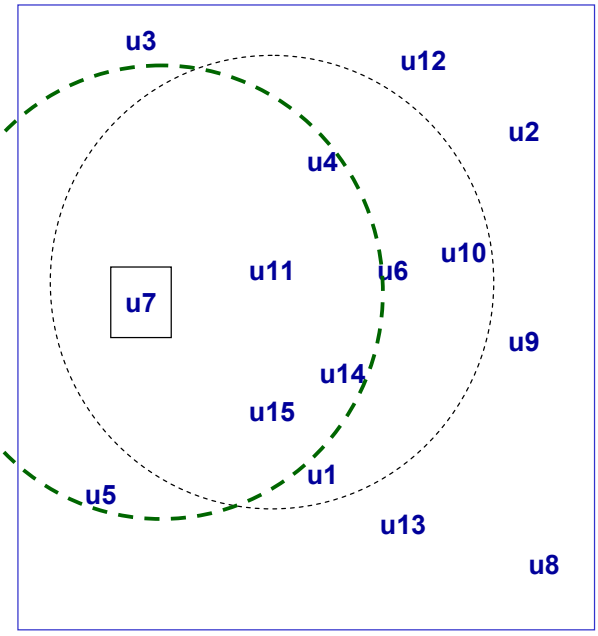


p=u11

median(d(p,u) | u in U}=**3.1**

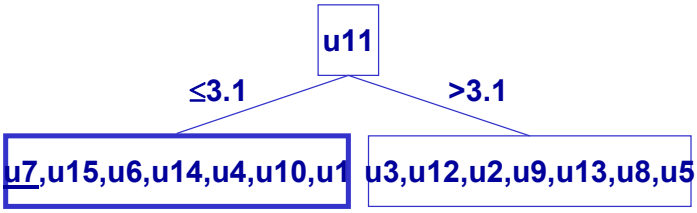


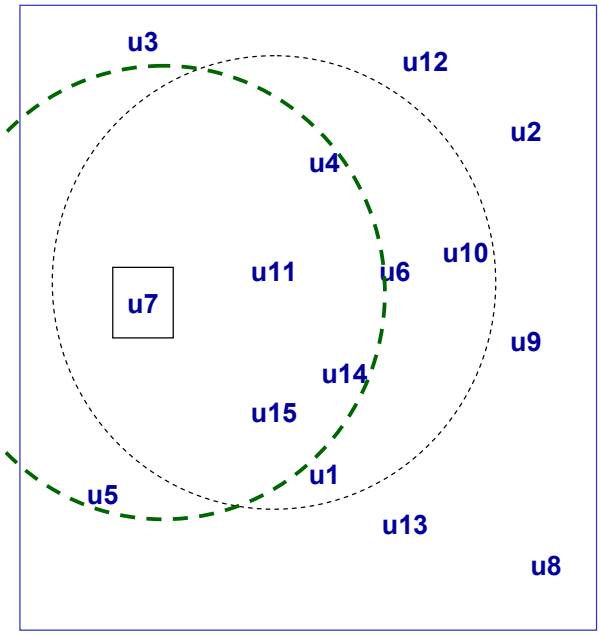
Επιλογή pivot



p=u7

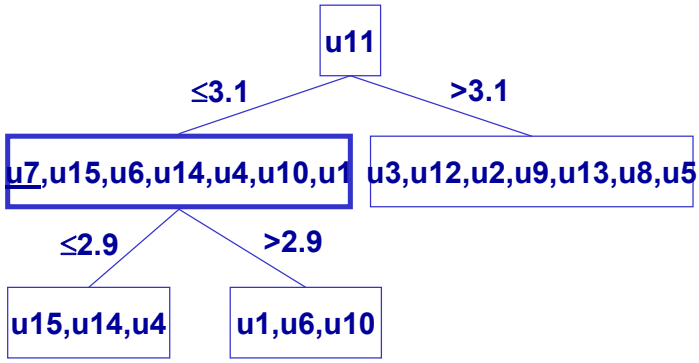
median(d(p,u) | u in ()}=**2.9**

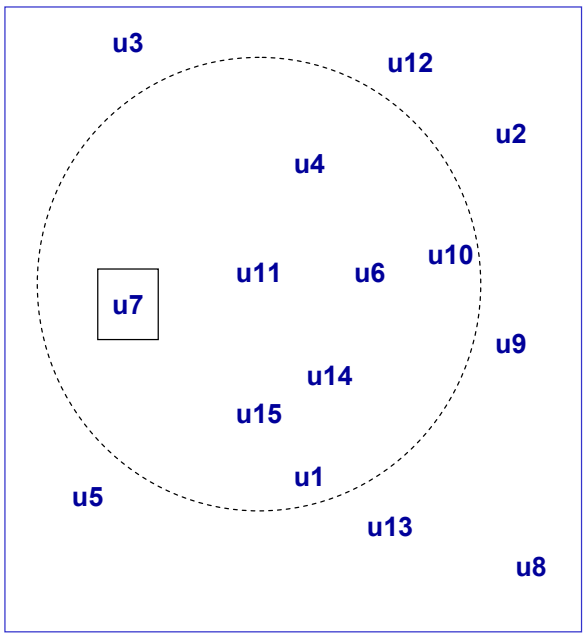




p=u7

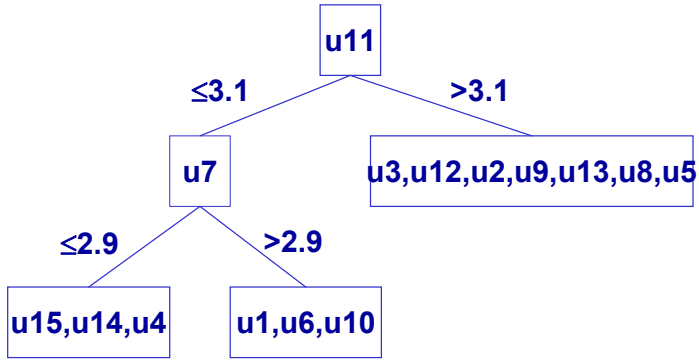
median(d(p,u) | u in ()}=**2.9**



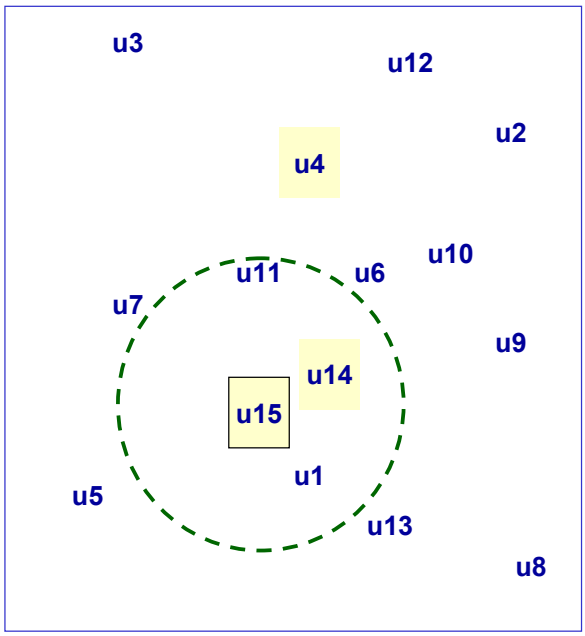


p=u7

median(d(p,u) | u in ()}=**2.9**

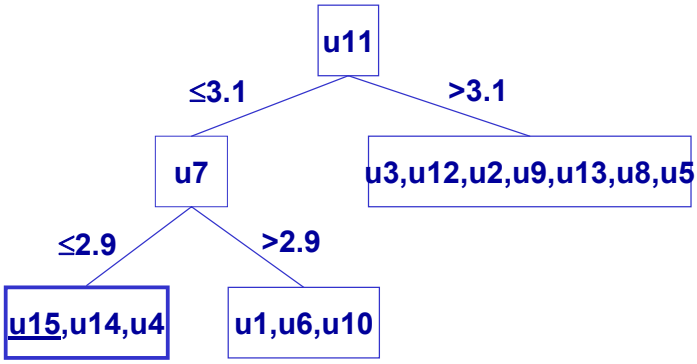


Επιλογή pivot

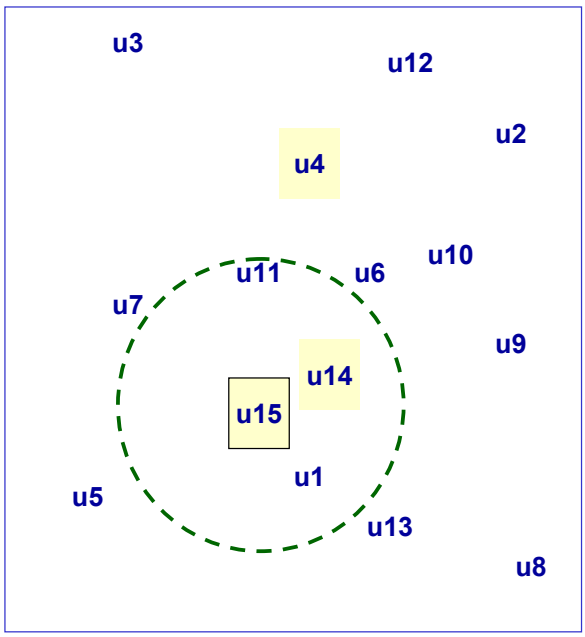


p=u15

median(d(u15,u14), d(u15,u4)) =**2.5**

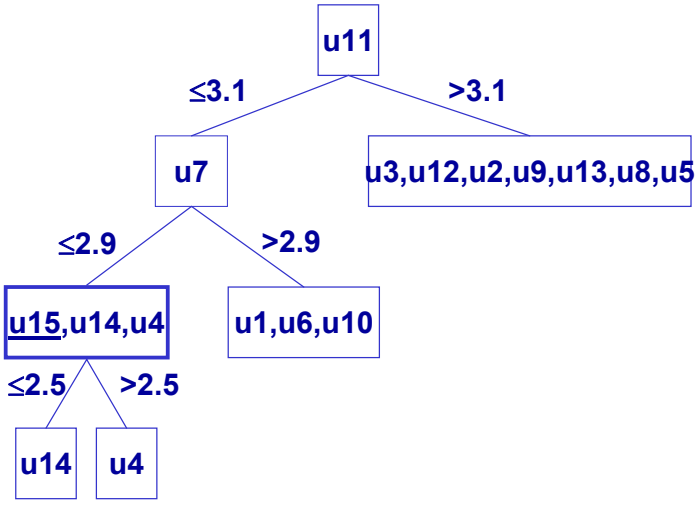


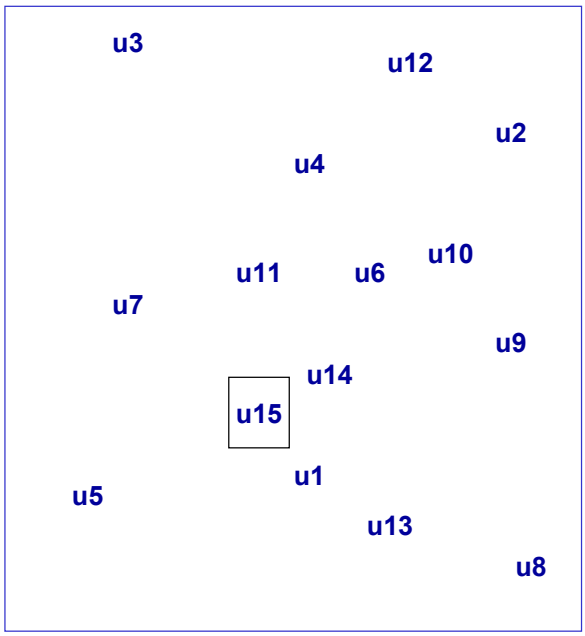
Διαμέριση



p=u15

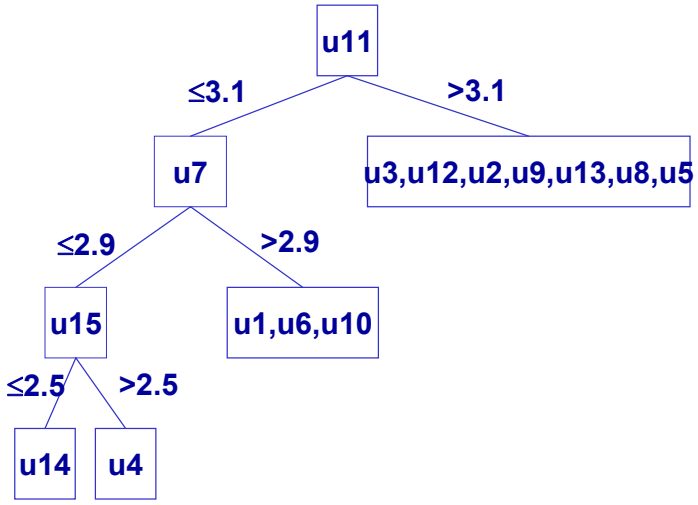
median(d(u15,u14), d(u15,u4)) =**2.5**



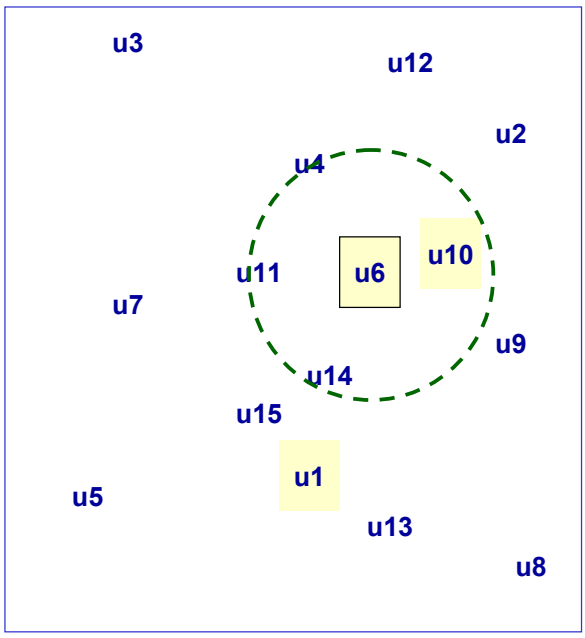


p=u15

median(d(u15,u14), d(u15,u4)) =**2.5**

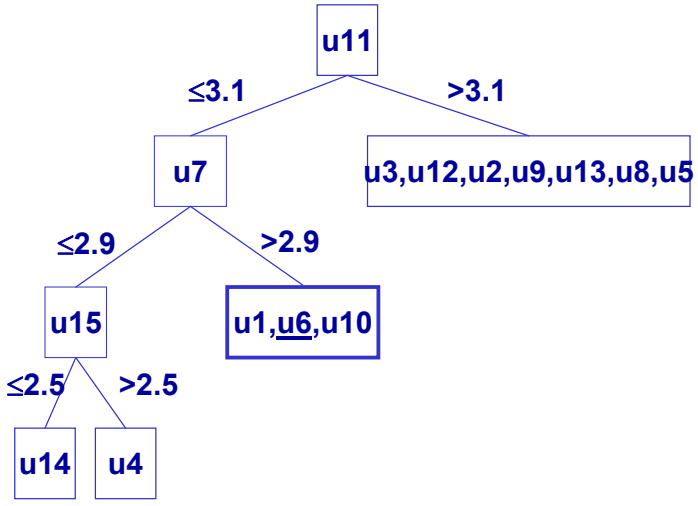


Επιλογή pivot

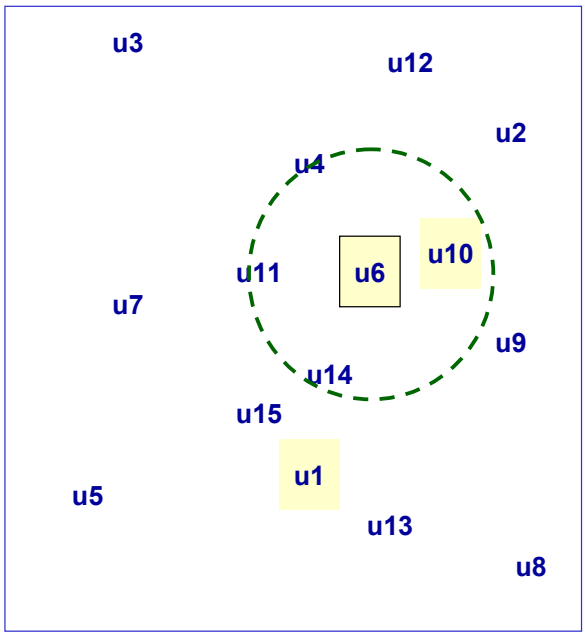


p=u6

median(d(u6,u1), d(u6,u10)) =**1.8**

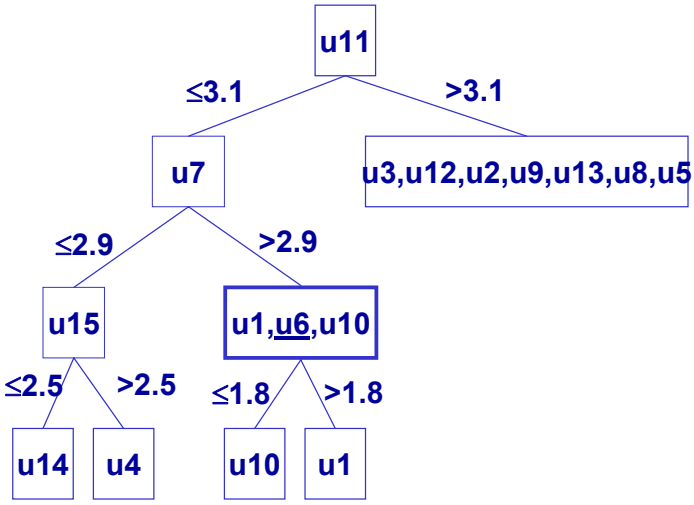


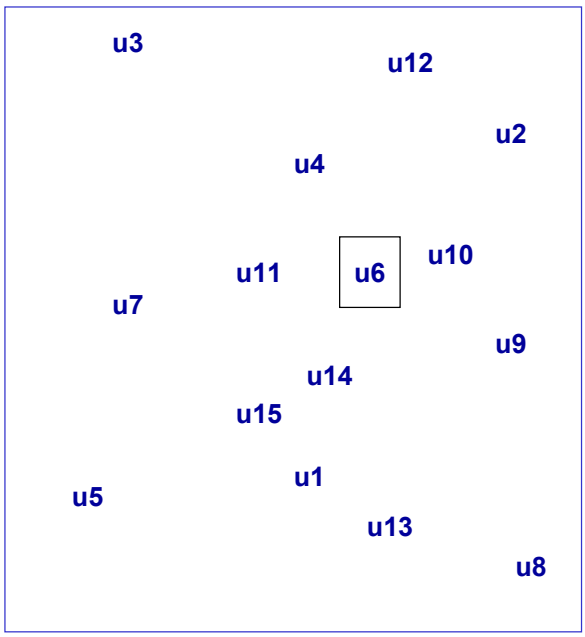
Διαμέριση



p=u6

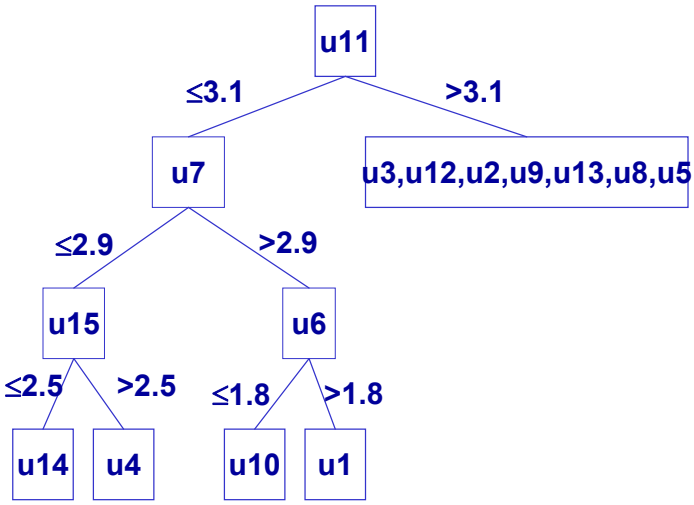
median(d(u6,u1), d(u6,u10)) =**1.8**



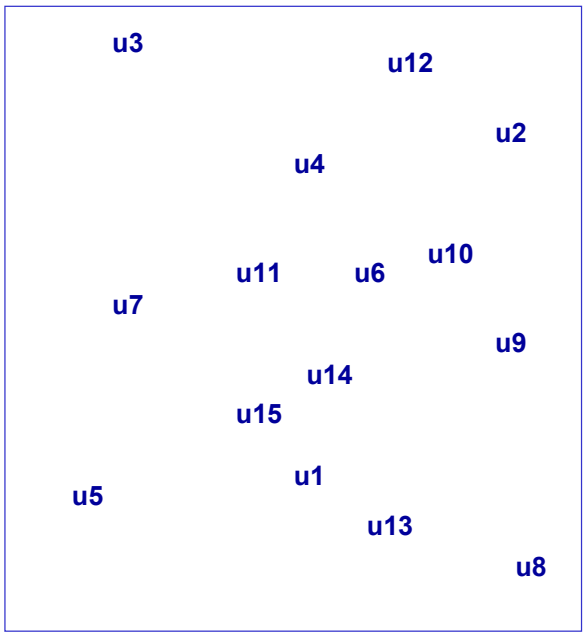


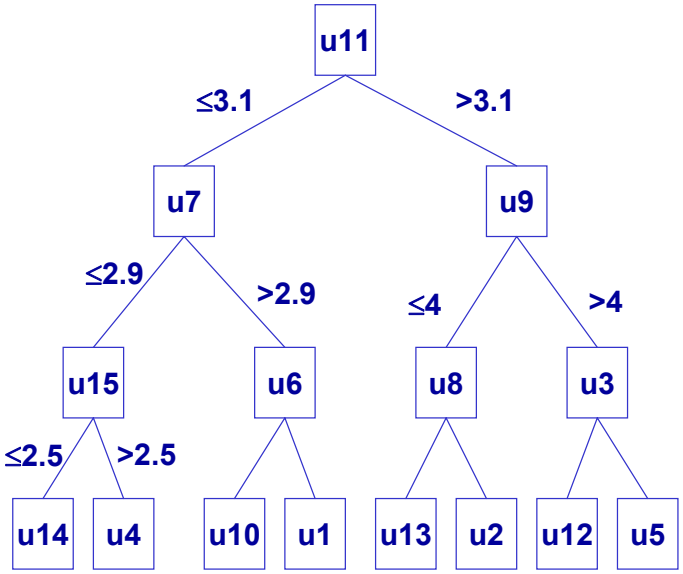
p=u6

median(d(u6,u1), d(u6,u10)) =**1.8**



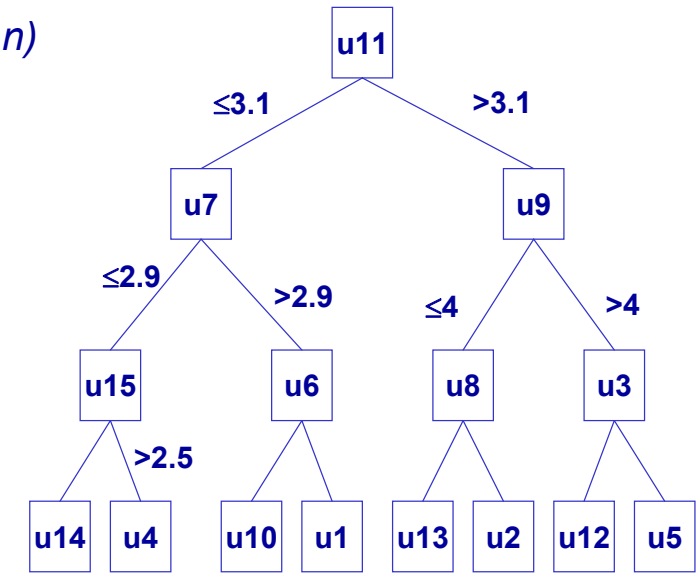
Το τελικό αποτέλεσμα





Vantage-point-trees

- Χώρος: *O(n)*
- *Χρόνος κατασκευής: O(n* log*n)* distance evaluations
  - (διότι είναι ισοζυγισμένα)



Vantage-point-trees: Αλγόριθμος Αναζήτησης

Έστω επερώτηση (Q,ε)

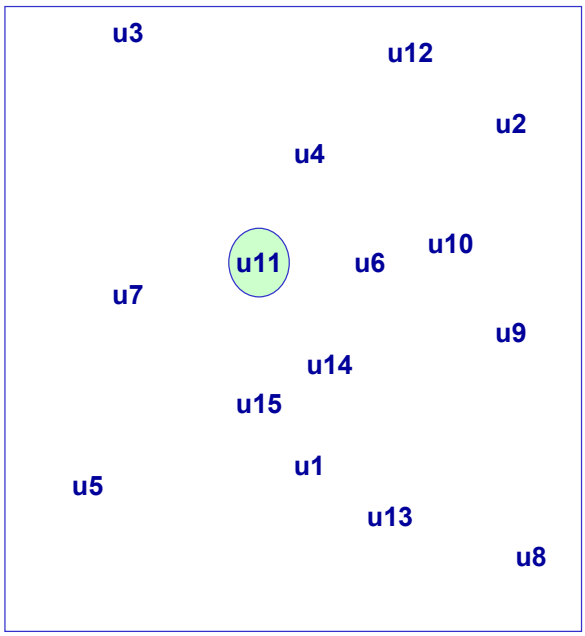
- 1/ Μετράμε την απόσταση του Q από το pivot p, δηλαδή d(Q,p)
- 2/ Αν d(Q,p)-ε <= M πάμε στο αριστερό υποδέντρο
  Αν d(Q,p) +ε > M πάμε στο δεξί υποδέντρο
  (ενδέχεται να μπούμε και στα δύο υποδένδρα)
- 3/ Επιστρέφουμε τα στοιχεία που έχουν απόσταση <= ε από το Q

Κόστος αναζήτησης:
O (log n) υπολογισμοί απόστασης αν το ε είναι μικρό

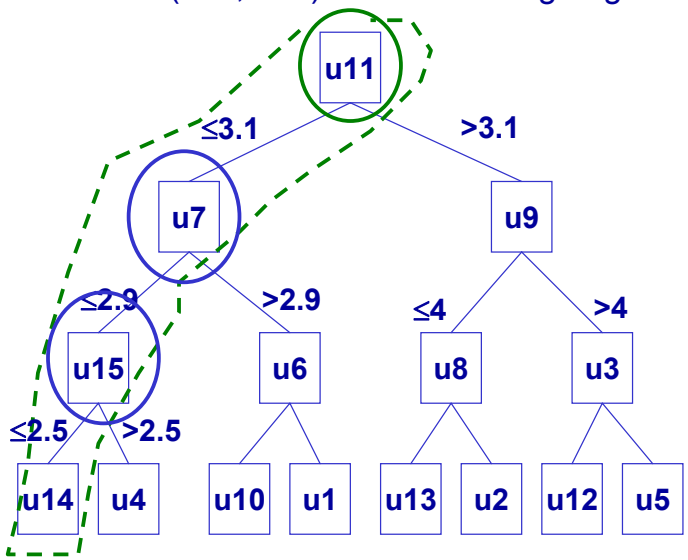
Q=(u11,0)

If d(Q,p)-ε <= M go left
if d(Q,p) +ε > M go right

If d(u11,u11)-0 =0 <= 3.1 **go left**
if d(u11,u11) +0 =0 > 3.1 go right



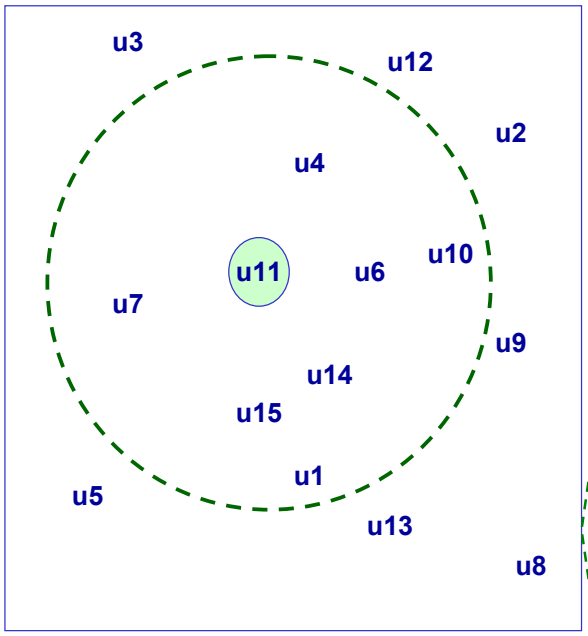
If d(u11,u7)-0 =1 <= 2.9 **go left**
if d(u11,u7) +0 =1 > 2.9 go right
If d(u11,u15)-0 =1.5 <= 2.5 **go left**
if d(u11, u15)+0 =1.5 > 2.5 go right



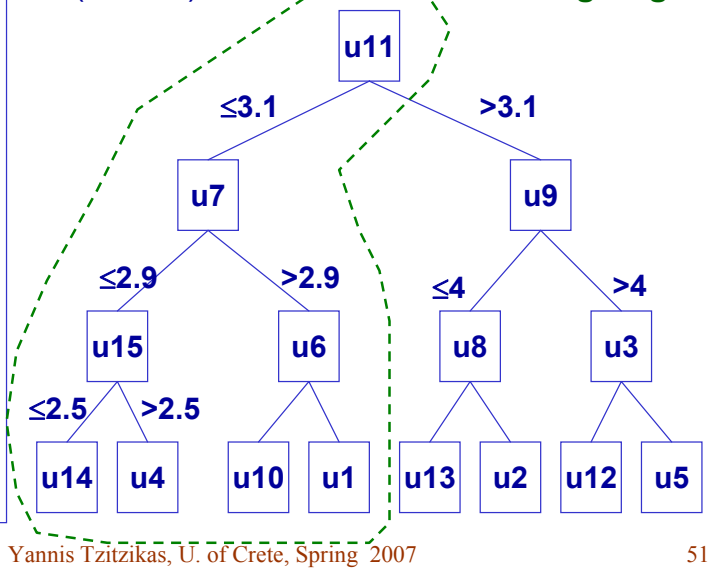
Q=(u11,3.1)

If d(Q,p)-ε <= M go left
if d(Q,p) +ε > M go right

If d(u11,u11)-3.1 =-3.1 <= 3.1 **go left**
if d(u11,u11) +3.1 =3.1 > 3.1 go right



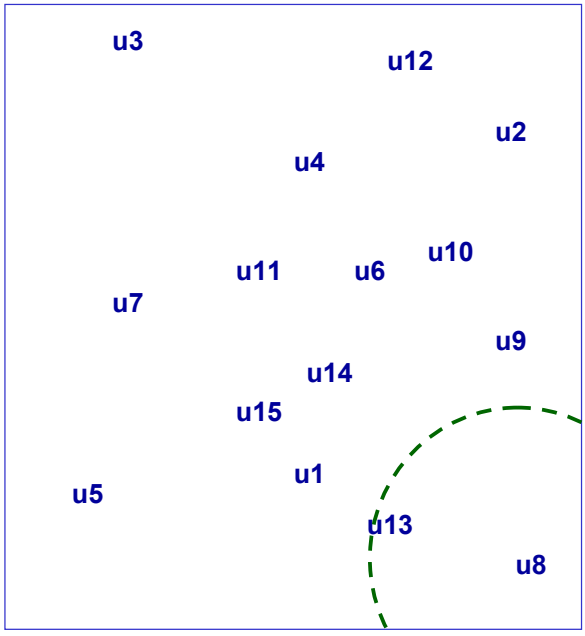
If d(u11,u7)-3.1 =2.1-3.1=-1 <= 2.9 **go left**
if d(u11,u7)+3.1=2.1 + 3.1 =5.1 >2.9 **go right**



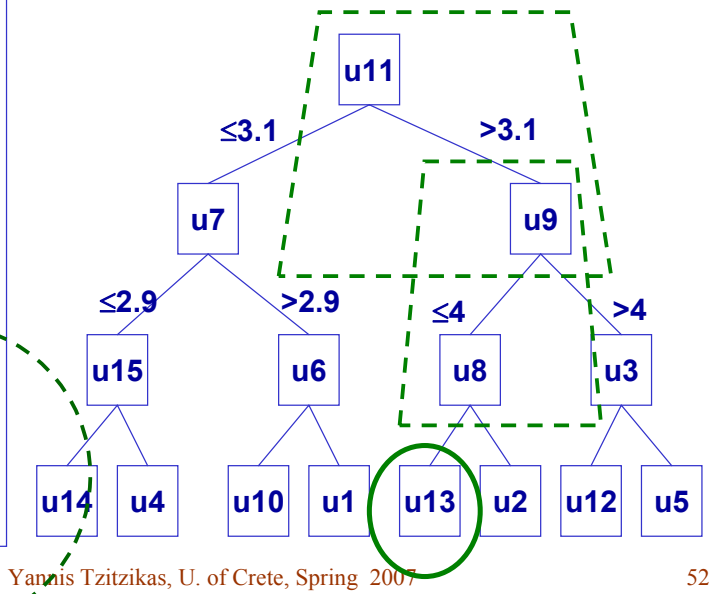
Q=(u8,1) // μικρό ε

If d(Q,p)-ε <= M go left
if d(Q,p) +ε > M go right

If d(u8,u11)-1=4.2-1=3.2<= 3.1 go left
if d(u8,u11)+1=4.2+1=5.2 >3.1 **go right**



If d(u8,u9)-1=2-1=1 <= 4 **go left**
if d(u8,u9)+1=2+1=2 > 4 go right

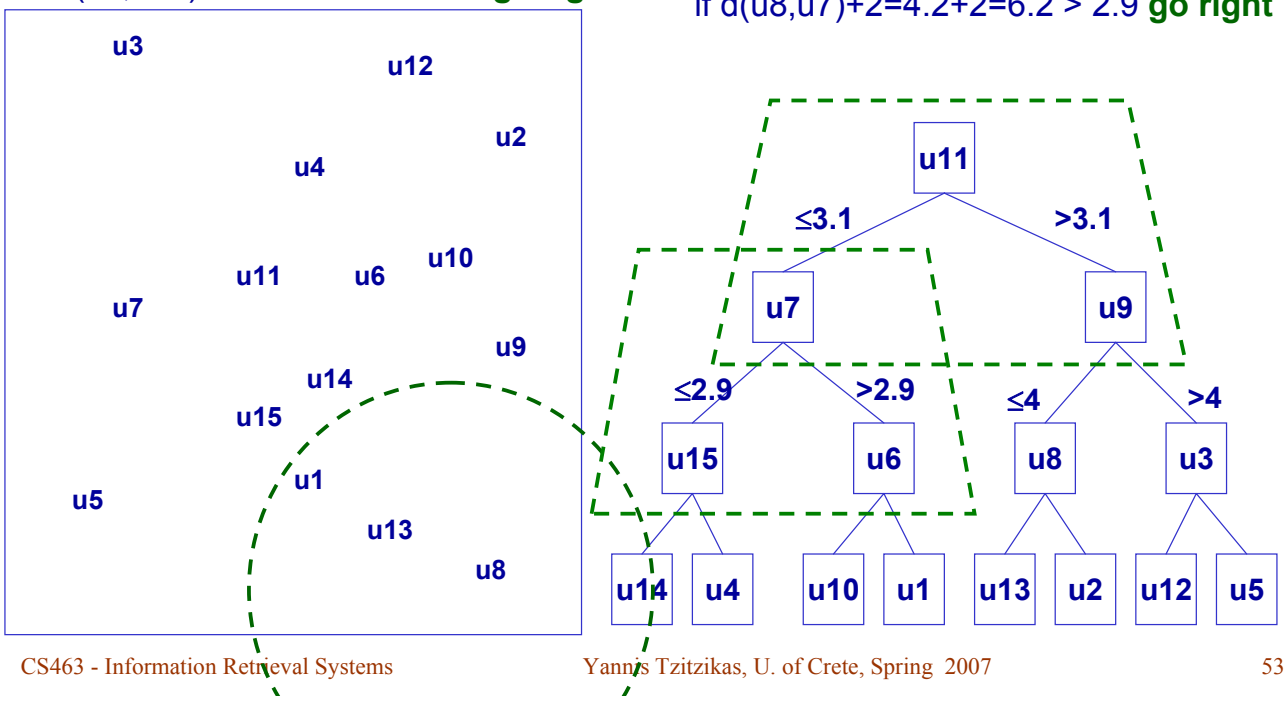


Q=(u8,2) // μεγάλο ε

If d(Q,p)-ε <= M go left
if d(Q,p) +ε > M go right

If d(u8,u11)-2=4.2-2=2.2<= 3.1 **go left**
if d(u8,u11)+2=4.2+2=6.2 >3.1 **go right**

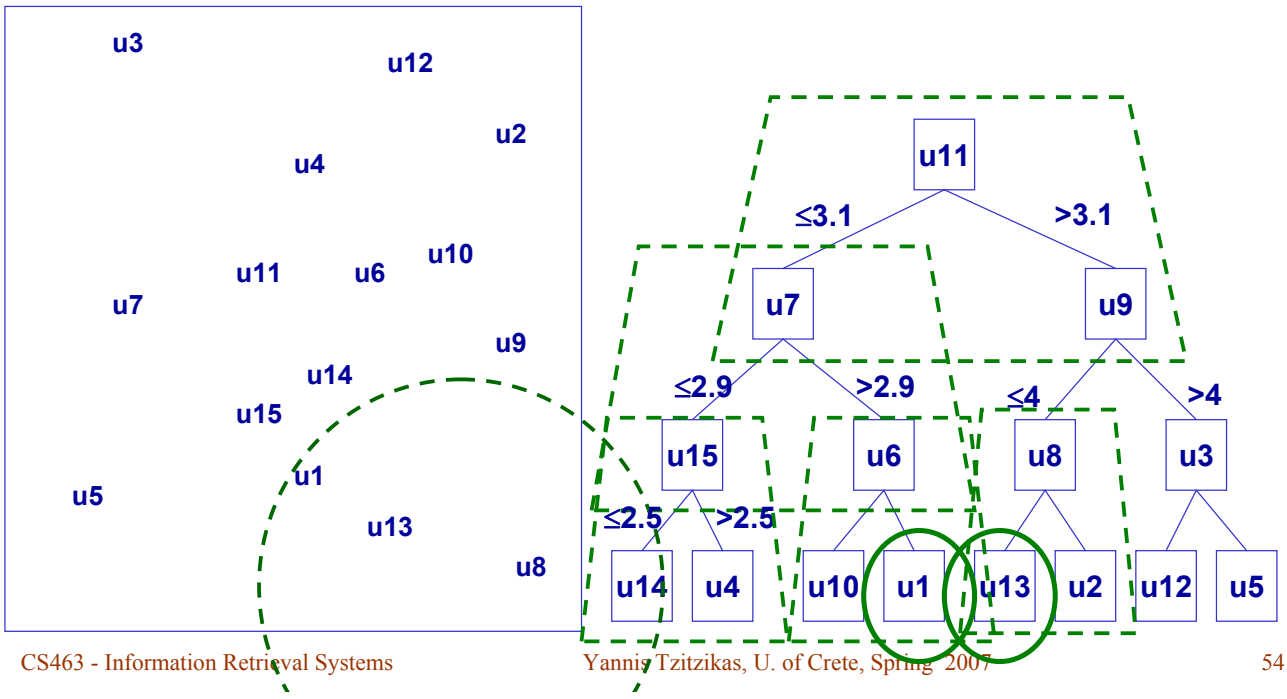
If d(u8,u7)-2=4.2-2=2.2 <= 2.9 **go left**
if d(u8,u7)+2=4.2+2=6.2 > 2.9 **go right**



Q=(u8,2)

If d(Q,p)-ε <= M go left
if d(Q,p) +ε > M go right

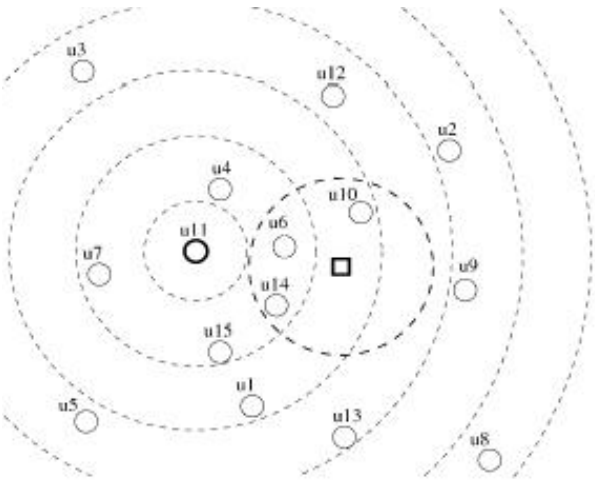
If d(u8,u15)-2=3-2=1<= 2.5 **go left**
if d(u8,u15)+2=3+2=5 >2.5 **go right**

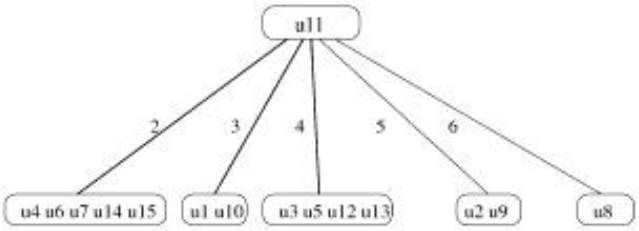


BKT (**Burkhard-Keller Tree**)

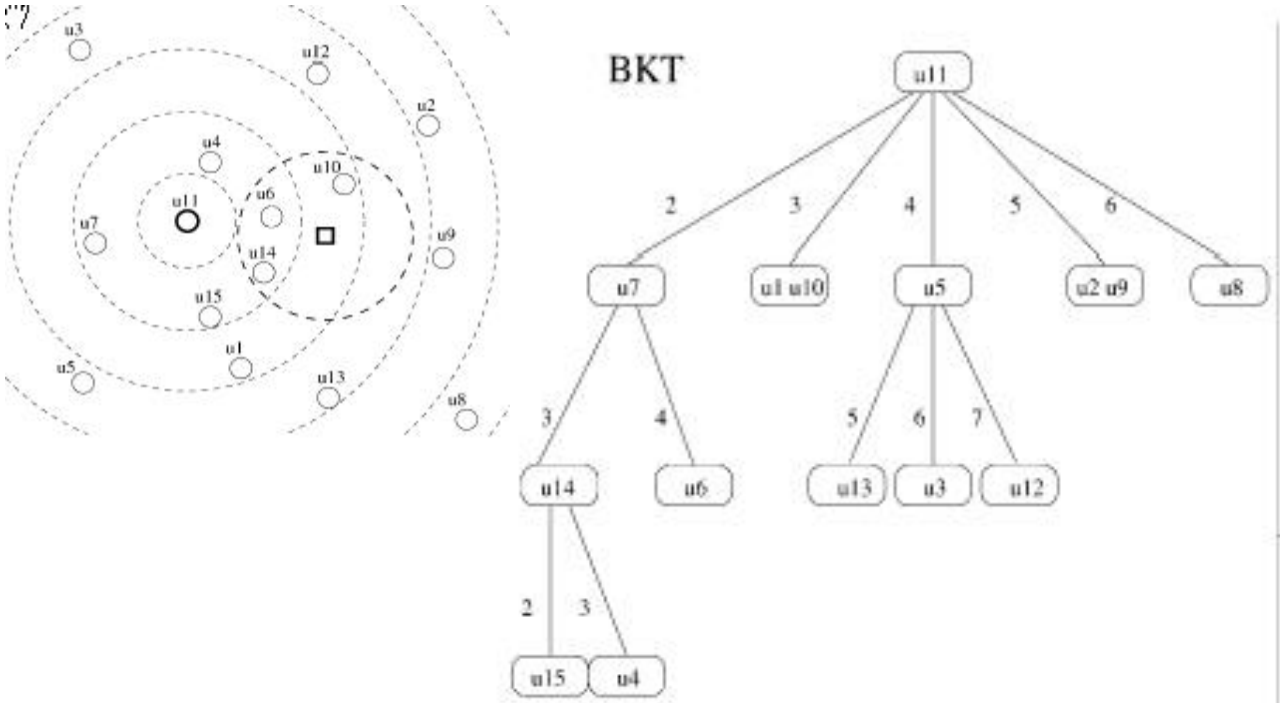
- An arbitrary element *p* in *U* is selected as the root of the tree.
- For each distance *i>0*, we define U_i={*u in U* | *d(u,p)=i*}
  - the set of all the elements at distance *i* to the root *p*.
- Then, for any nonempty U_i, we build a child of *p* (labelled *i*), and we recursively build the BKT for U_i.
- This process can be repeated until there is only one element to process, or until there are no more than *b* elements (and we store a *bucket* of size *b*). All the elements selected as roots of subtrees are called *pivots*.

BKT: Το πρώτο επίπεδο





BKT: Ολόκληρο



BKT: Αλγόριθμος Αναζήτησης

Είσοδος: Συλλογή C, Επερώτηση *Q,ε*

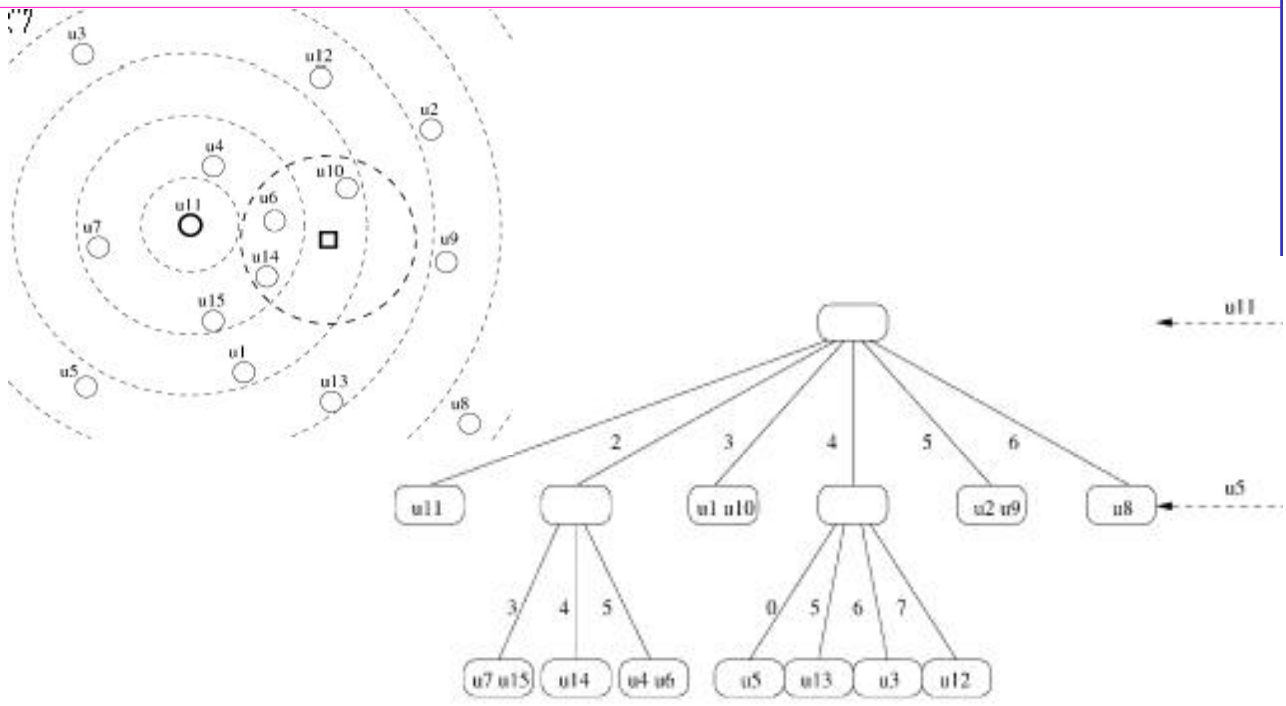
Έξοδος: *ans(Q,ε)*

- Ξεκίνα από τη ρίζα
- Μπες σε όλα τα παιδιά i τ.ω. d(Q,p)-ε <= i <= d(Q,p) +ε
- Προχώρησε αναδρομικά
- Όταν φτάσουμε σε ένα φύλλο, ελέγχουμε σειριακά όλα τα στοιχεία του

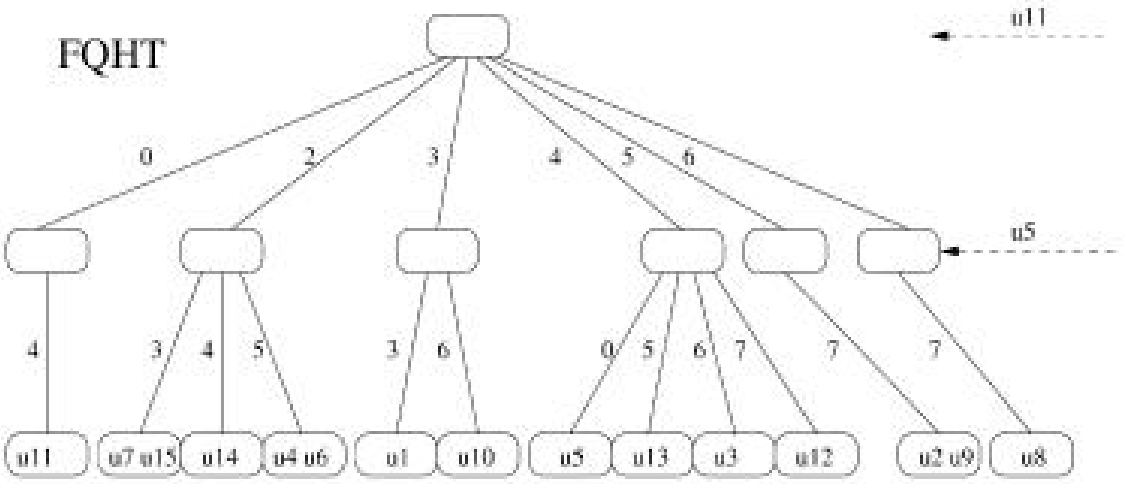
Fixed Query Tree (FQT) [Baeza-Yates 94]

- Διαφορές με το BKT:
  - Στο BKT κάθε κόμβος έχει το δικό του pivot
  - Στο FQT όλοι οι κόμβοι ενός επιπέδου έχουν το ίδιο pivot
  - Όλα τα στοιχεία αποθηκεύονται στα φύλλα
- Κόστος κατασκευής:
  - O (n log n) distance evaluations
- Κόστος αναζήτησης
  - O(n^a), 0<a<1

Fixed Query Tree (FQT)



FQHT



(C) not tree-based indexes.
AESA(Approximating Eliminating Search Algorithm)

- Βασίζεται σε έναν πίνακα με n(n-1)/2 προϋπολογισμένες αποστάσεις

$$\begin{array}{l} u_1 \\ u_2 \quad D(2,1) \\ u_3 \quad D(3,1) \quad D(3,2) \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ u_n \quad D(n,1) \; D(n,2) \; \ldots \quad D(n,n-1) \\ \quad\; u_1 \quad\quad u_2 \quad \ldots . \; u_{n-1} \quad u_n \end{array}$$

AESA: Αλγόριθμος Αναζήτησης

Έστω επερώτηση (Q,ε)

1/ Επιλέγουμε τυχαία ένα σημείο p

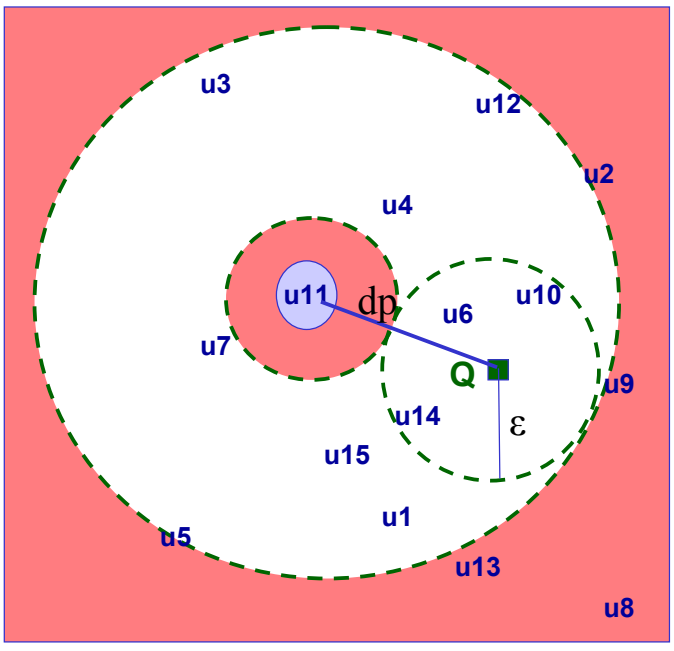
2/ Μετράμε την απόσταση του Q από το pivot p, dp := d(Q,p)

3/ Απέκλεισε όλα τα σημεία u που δεν ικανοποιούν το:

dp - ε <= D(u,p) <= dp + ε // βάσει του πίνακα που ήδη έχω

4/ Συνέχισε έτσι έως ότου δεν έχουν απομείνει άλλα σημεία

AESA: Αλγόριθμος Αναζήτησης: Παράδειγμα

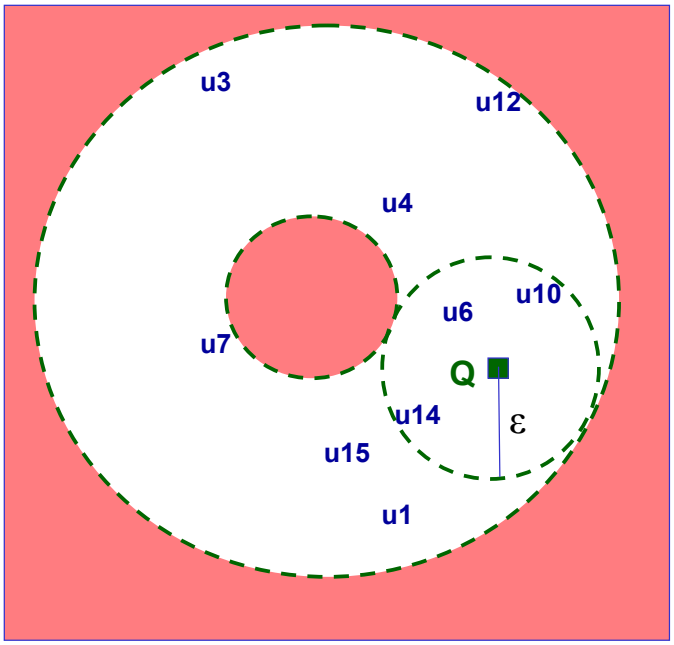


p=u11

Αποκλεισμός των u που δεν ικανοποιούν:

dp - ε <= D(u,p) <= dp + ε

AESA: Αλγόριθμος Αναζήτησης: Παράδειγμα



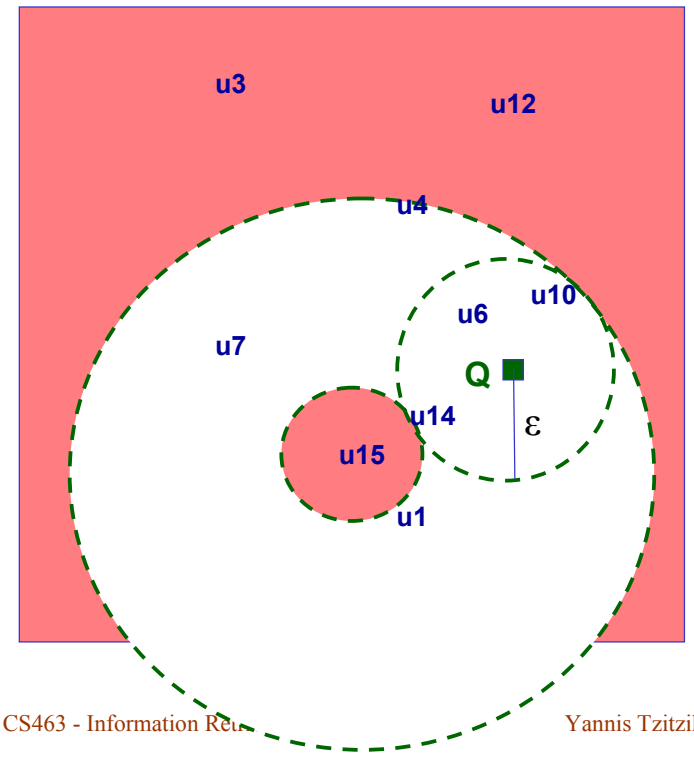
p=u11

Αποκλεισμός των u που δεν ικανοποιούν:

$dp - \varepsilon <= D(u,p) <= dp + \varepsilon$

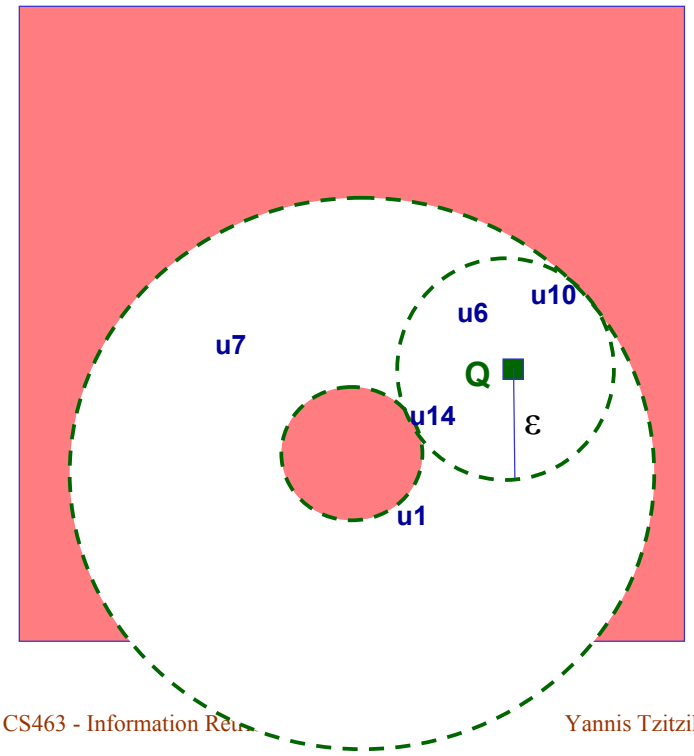
Επόμενο pivot p=u15

AESA: Αλγόριθμος Αναζήτησης: Παράδειγμα



p=u15

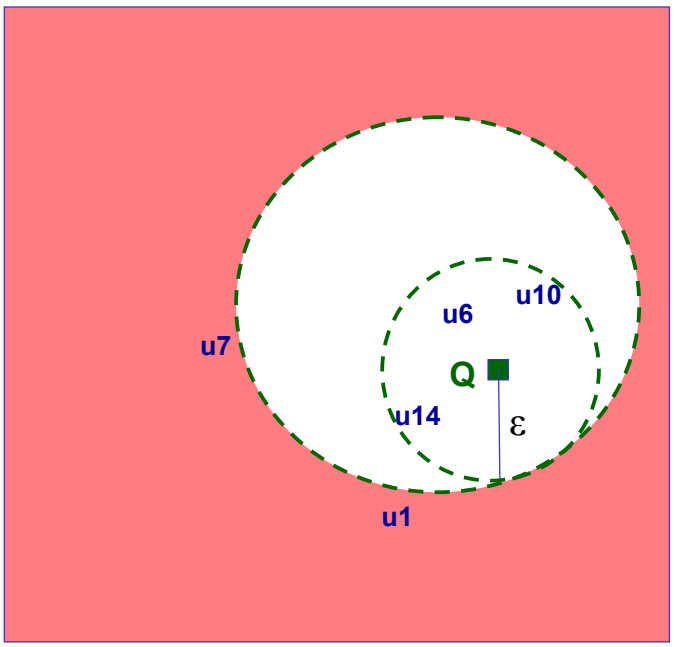
AESA: Αλγόριθμος Αναζήτησης: Παράδειγμα



p=u15

Επόμενο pivot p=u6

AESA: Αλγόριθμος Αναζήτησης: Παράδειγμα

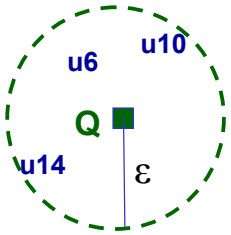


p=u6

Αποκλεισμός των u που δεν ικ.:

$D(u,p) <= dp + \varepsilon$

AESA: Αλγόριθμος Αναζήτησης: Παράδειγμα



AESA
(Approximating Eliminating Search Algorithm)

- Χώρος: O(n^2) (το οποίο είναι πολύ μεγάλο)
- Χρόνος κατασκευής: O(n^2)
- Experimental query time: O(1)

Διάρθρωση Διάλεξης

- Το Πρόβλημα, Εφαρμογές και Παραδείγματα
- Επερωτήσεις Ομοιότητας

- Feature-based Indexing and Retrieval
  - Μέθοδοι Χωρικής Πρόσβασης
  - Case Study: 1D Time Series
  - Case Study: 2D Images

- Ανάκτηση Πολυμέσων χωρίς τη χρήση Features
  - μετρικά ευρετήρια

Αναφορές

- Modern Information Retrieval, Chapter 12
- Edgar Chavez, Gonzalo Navaro, Ricardo Baeza-Yates, Jose Luis Marroquin, Searching in metric spaces, *ACM Computing Surveys*, 33(3), 273-321, September 2001

- [Christian Bohn, Stefan Berchtold, Daniel Keim, Searching in multidimensional spaces, *ACM Computing Surveys*, 33(3) , 322-373, September 2001]