Fondements de la recherche par similarité

Algorithmes et structures pour la recherche par similarité: Séance 1

Alexis Joly
http://www-rocq.inria.fr/ ajoly/

INRIA Rocquencourt, Equipe-projet IMEDIA

April 25, 2008

Plan

Plan

Plan

Plan

Plan

Chapitre I

Introduction

Remerciements

Michel Crucianu (CNAM)
http://www-rocq.inria.fr/ crucianu/
Yury Lifshits (CIT) http://yury.name/
The Homepage of Nearest Neighbors and Similarity Search
http://simsearch.yury.name/

Objectif

Prétraiter une base de n objets
pour qu'étant donné un objet requête,
on puisse rapidement déterminer
les plus proches voisins dans la base

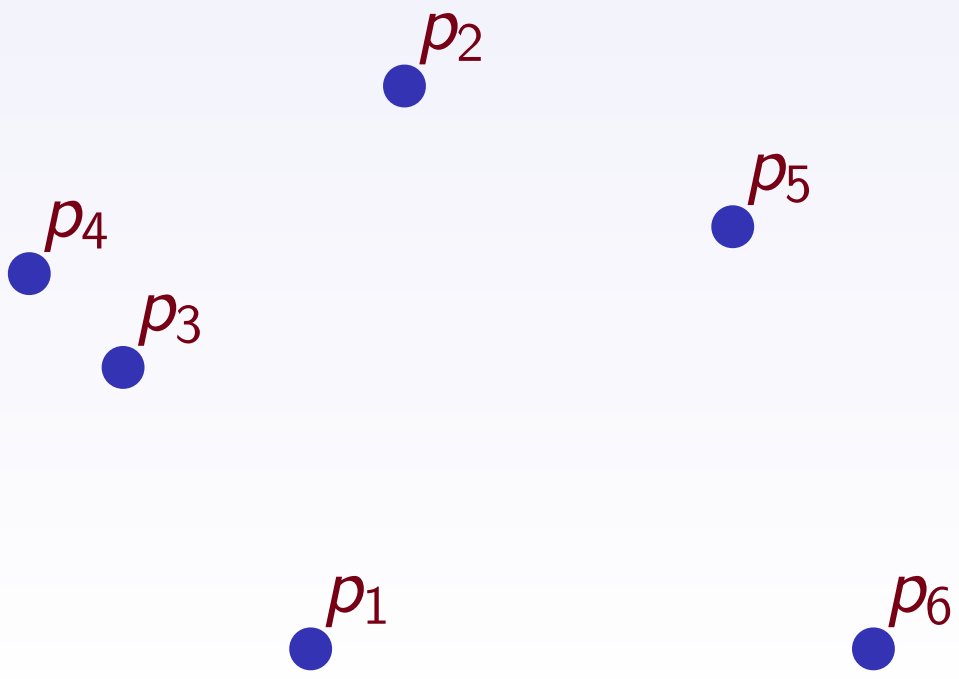
Plus concrètement

Espace de recherche: Espace $\mathbb{U}$, fonction similarité σ

Entrée: database $S = \{p_1, \ldots, p_n\} \subseteq \mathbb{U}$

Requête: $q \in \mathbb{U}$

Tâche: trouver $\text{argmax}_{p_i}\ \sigma(p_i, q)$



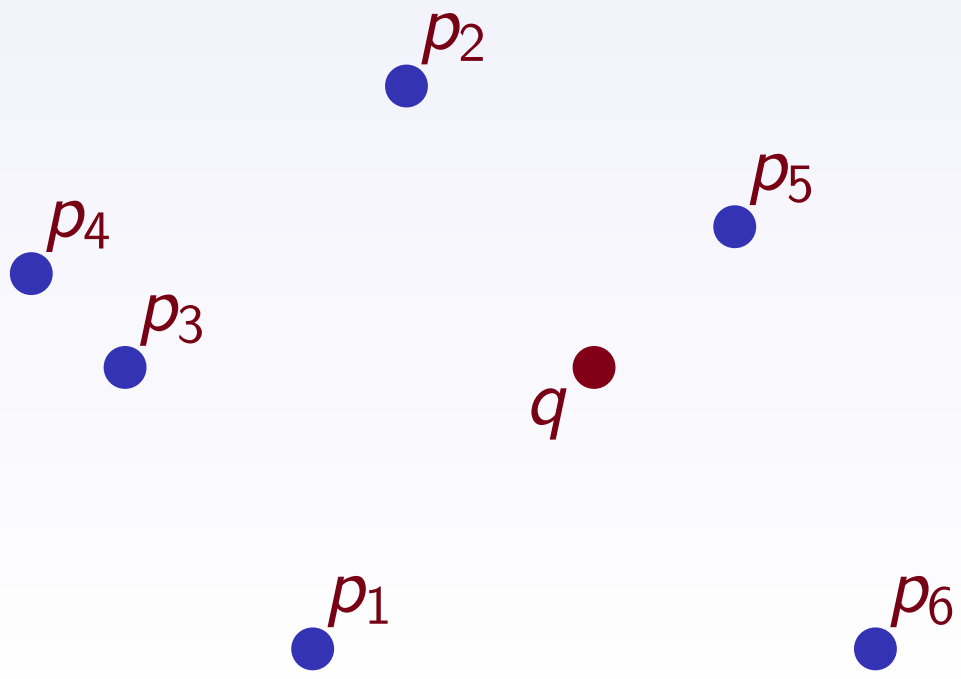
Plus concrètement

Espace de recherche: Espace $\mathbb{U}$, fonction similarité σ

Entrée: database $S = \{p_1, \ldots, p_n\} \subseteq \mathbb{U}$

Requête: $q \in \mathbb{U}$

Tâche: trouver $\text{argmax}_{p_i}\ \sigma(p_i, q)$



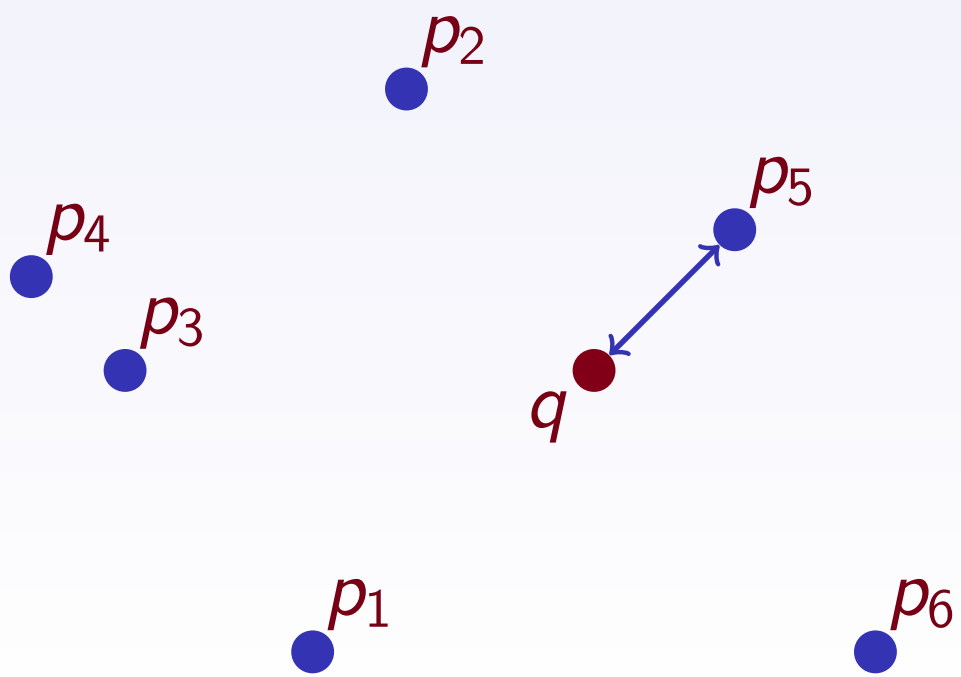
Plus concrètement

Espace de recherche: Espace $\mathbb{U}$, fonction similarité σ

Entrée: database $S = \{p_1, \ldots, p_n\} \subseteq \mathbb{U}$

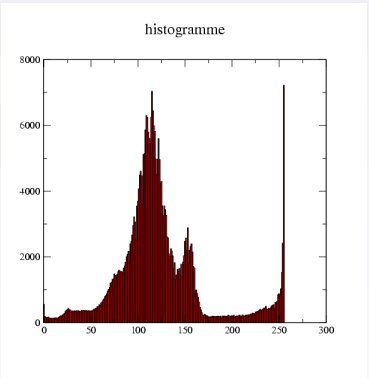
Requête: $q \in \mathbb{U}$

Tâche: trouver $\text{argmax}_{p_i}\ \sigma(p_i, q)$



Exemple d'application: Recherche par le contenu visuel

- Cas d'école: Histogramme HSV



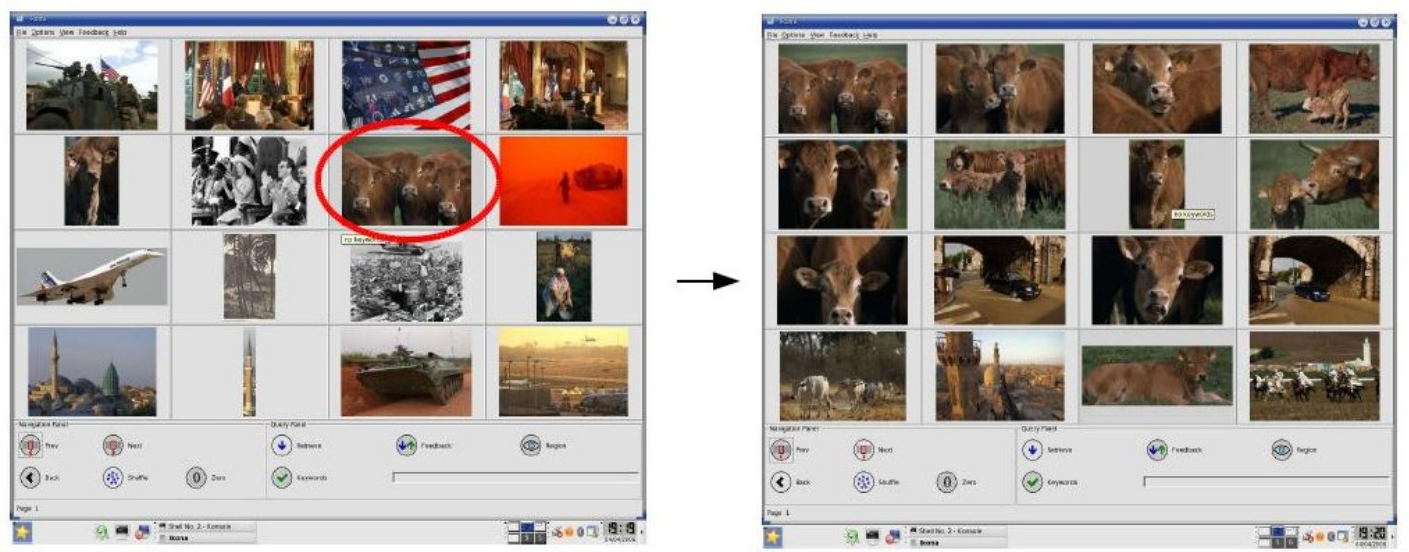
$$\mathbf{X} = (x_1, \ldots, x_D)$$

- Intersection d'histogrammes:

$$\sigma(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^{D} min(x_i, y_i)}{min\left(\sum_{i=1}^{D} x_i, \sum_{i=1}^{D} y_i\right)}$$

Exemple d'application: Recherche par le contenu visuel

- Recherche d'images par le contenu (requête exemple)

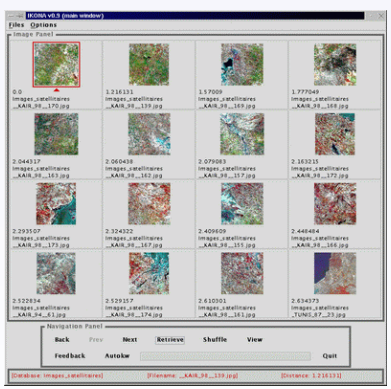
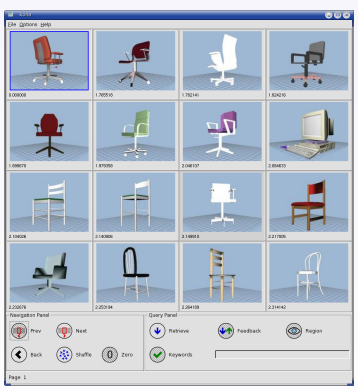


- Utilisation du contenu de l'image uniquement

Exemple d'application: Recherche par le contenu visuel

- Autres types de base de données



Applications (1/5) Recherche d'information

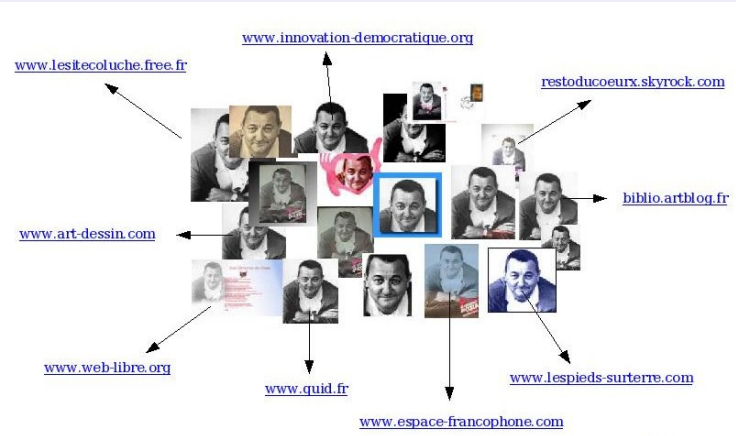
- Recherche par le contenu (images médicales, video, musiques, séries temporelles, textes)
- Correction orthographique
- Bases géographiques (pb du bureau de poste)
- Recherche de séquences ADN
- Recherche de pages web similaires
- Recherche sémantique, matching de concepts

Applications (2/5) Machine Learning

- classifieur kNN: classifie par la majorité des k plus proches voisins dans un ensemble d'apprentissage. E.g. reconnaissance de visages, empreintes, reconnaissance de locuteur, optical characters
- Interpolation (Voronoi diagram)

Applications (3/5) Fouille de données

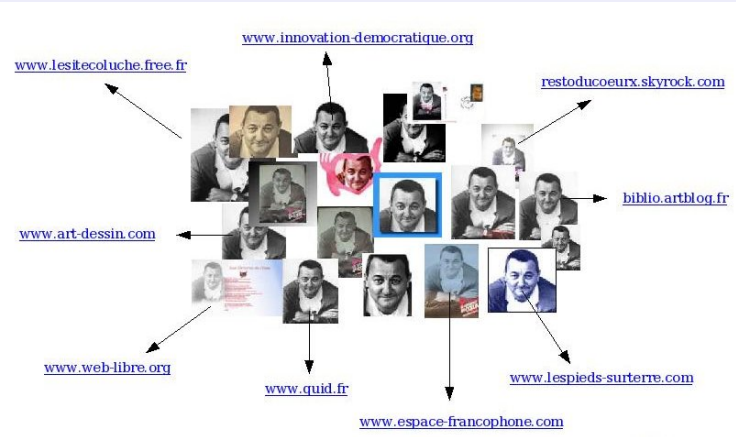
- Détection de copies ou d'images presque indentiques sur le web



- Calcul de co-occurrences par similarité (détection de synonymes, traduction automatique...)

Applications (3/5) Fouille de données

- Détection de copies ou d'images presque indentiques sur le web



- Calcul de co-occurrences par similarité (détection de synonymes, traduction automatique...)

Différence clé:

Problèmes hors-ligne (NN précalculés)

Applications (4/5) Problèmes Biparti

- Systèmes de recommendation (recherche de films pertinents par rapport à un ensemble de documents déjà vus)
- Ciblage comportemental (pub la plus pertinente pour un utilisateur donné)

Applications (4/5) Problèmes Biparti

- Systèmes de recommendation (recherche de films pertinents par rapport à un ensemble de documents déjà vus)
- Ciblage comportemental (pub la plus pertinente pour un utilisateur donné)

Différences clé:

Requête et base peuvent avoir des natures différentes
Similarités définies par des connexions

Applications (5/5) Comme brique de base

- Quantification vectorielle
- Compression MPEG (recherche de fragments similaires)
- Regroupement (Clustering)

Variations du problème

Type de solution requise:

- Recherche exact ou approximative
- Dynamicité: objets qui bougent, , insertions/suppressions, changements de mesures de similarité

Variations du problème

Type de solution requise:

- Recherche exact ou approximative
- Dynamicité: objets qui bougent, , insertions/suppressions, changements de mesures de similarité

Types de requête:

- Plus proche voisin: musée le plus proche de mon hôtel
- Plus proche voisin inversé: tous les musées pour lesquels mon hôtel est le plus proche
- Range queries: Tous les musées dans un rayon de 2km autours de mon hôtel
- Paire la plus rapprochée: Musée et hôtel les plus rapprochés
- Jointure spatiale: paires d'hôtel et musée distants de moins d'1 km
- Plus proches voisins multiples: musées les plus proches d'un ensemble d'hôtel

Bref Historique

1908 diagramme de Voronoi

Bref Historique

1908 diagramme de Voronoi

1967 classifieur kNN par Cover et Hart

Bref Historique

1908 diagramme de Voronoi

1967 classifieur kNN par Cover et Hart

1973 Problème du bureau de poste posé par Knuth

Bref Historique

1908 diagramme de Voronoi

1967 classifieur kNN par Cover et Hart

1973 Problème du bureau de poste posé par Knuth

1988 R-tree par Guttman

Bref Historique

1908 diagramme de Voronoi

1967 classifieur kNN par Cover et Hart

1973 Problème du bureau de poste posé par Knuth

1988 R-tree par Guttman

1997 M-tree par Ciaccia, Patella, Zezula

Bref Historique

1908 diagramme de Voronoi

1967 classifieur kNN par Cover et Hart

1973 Problème du bureau de poste posé par Knuth

1988 R-tree par Guttman

1997 M-tree par Ciaccia, Patella, Zezula

1999 LSH par Indyk

Bref Historique

1908 diagramme de Voronoi

1967 classifieur kNN par Cover et Hart

1973 Problème du bureau de poste posé par Knuth

1988 R-tree par Guttman

1997 M-tree par Ciaccia, Patella, Zezula

1999 LSH par Indyk

2006 "Similarity Search book" par Zezula, Amato, Dohnal and Batko

Bref Historique

1908 diagramme de Voronoi

1967 classifieur kNN par Cover et Hart

1973 Problème du bureau de poste posé par Knuth

1988 R-tree par Guttman

1997 M-tree par Ciaccia, Patella, Zezula

1999 LSH par Indyk

2006 "Similarity Search book" par Zezula, Amato, Dohnal and Batko

2008 1er Workshop International

Plan du cours

Trois cours:

1. Branch-and-bound: structures de données basées sur des arbres pour les espaces Euclidiens et métriques

Plan du cours

Trois cours:

1. Branch-and-bound: structures de données basées sur des arbres pour les espaces Euclidiens et métriques
2. Etude du M-tree et autres utilisations de l'inégalité triangulaire: M-tree, Techniques à base de marches dans des graphes, Techniques matricielles

Plan du cours

Trois cours:

1. Branch-and-bound: structures de données basées sur des arbres pour les espaces Euclidiens et métriques
2. Etude du M-tree et autres utilisations de l'inégalité triangulaire: M-tree, Techniques à base de marches dans des graphes, Techniques matricielles
3. Techniques de recherche approximatives: Locality-sensitive hashing, projections aléatoires, SASH, Dimension intrinsèques

Chapitre II

Méthodologie Branch and Bound

Espace métriques

Définition des espaces métriques

Espace métriques

Définition des espaces métriques

$M = (\mathbb{U}, d)$, fonction distance d satisfaisant:

Non negativité: $\forall s, t \in \mathbb{U} : \quad d(s,t) \geq 0$
Symmetrie: $\forall s, t \in \mathbb{U} : \quad d(s,t) = d(t,s)$
Identité: $d(s,t) = 0 \Rightarrow s = t$
Inégalité triangulaire:
$\forall r, s, t \in \mathbb{U} : \quad d(r,t) \leq d(r,s) + d(s,t)$

Espace métriques: Exemples basiques

- Espace métrique arbitraire, accès à un "oracle" pour la fonction distance
- Espace Euclidien k-dimensionnel avec distance Euclidienne,distance Euclidienne pondérée, distance de Manhattan, ou autres métriques L_p
- Chaînes de caractères avec distance de Hamming ou de Levenshtein (edit)

Espace métriques: Encore plus d'exemples

- Ensembles finis avec métrique de Jaccard
  $d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$
- Chemin le plus court dans un graphe connecté
- Ensembles avec distance de Hausdorff, ou Earth Mover Distance (EMD)

Espace métriques: Encore plus d'exemples

- Ensembles finis avec métrique de Jaccard
  $d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$
- Chemin le plus court dans un graphe connecté
- Ensembles avec distance de Hausdorff, ou Earth Mover Distance (EMD)

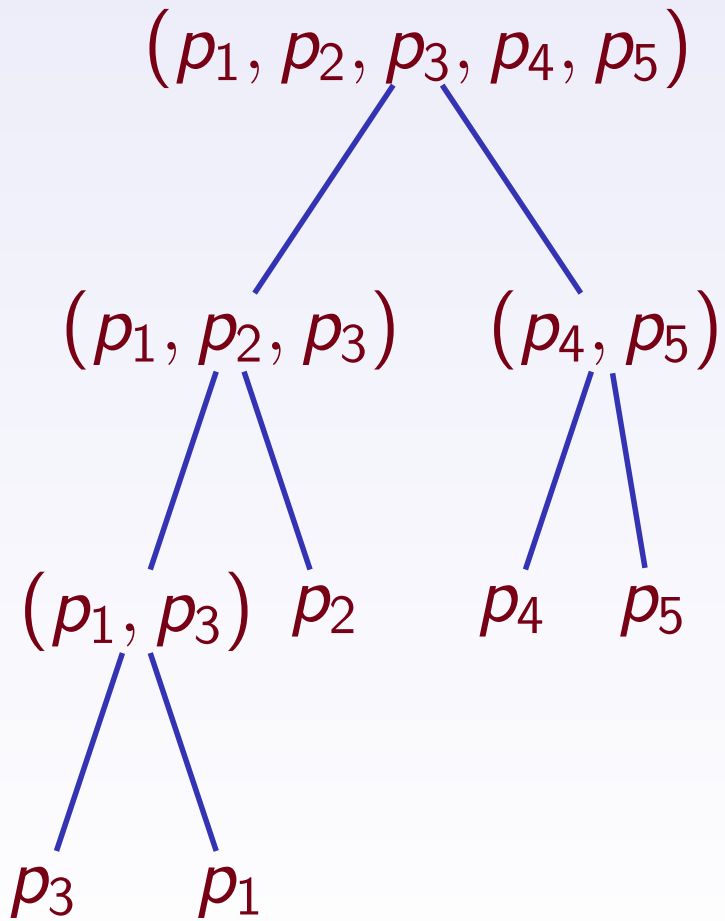
Espaces non-métriques (pas d'inégalité triangulaire):

- Produit scalaire entre vecteurs multidimensionnels
- Vote, matching (e.g. nombre de régions communes entre deux images)
- Graphes, réseaux (nombre d'amis communs dans un réseau social)

Branch and Bound: Recherche hiérarchique

Base $S = \{p_1, \ldots, p_n\}$
représentée par un arbre:

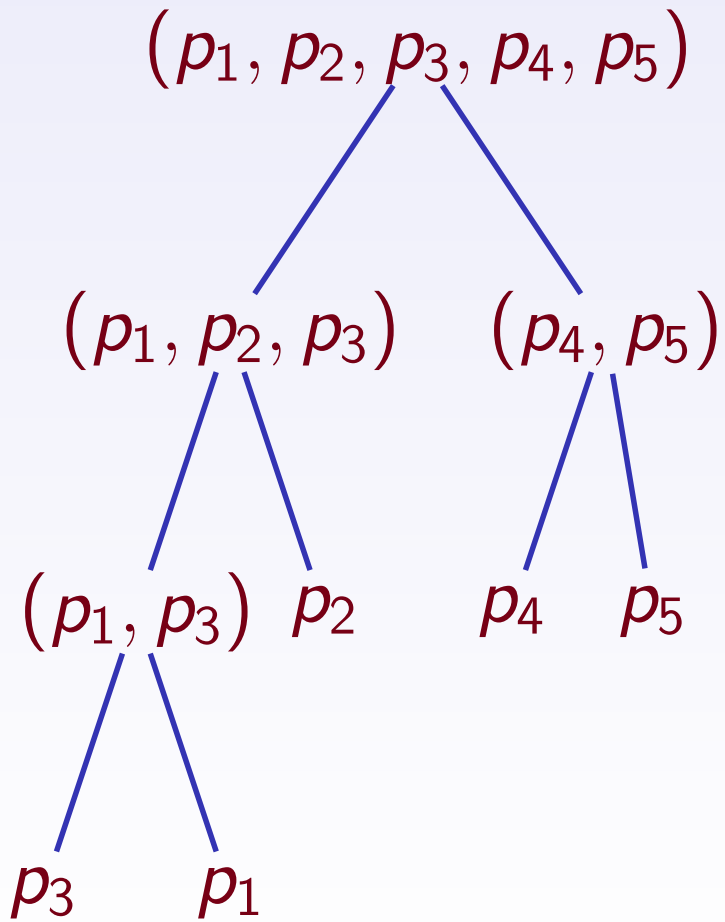
- Chaque noeud est un sous-ensemble de S
- La racine est S entier
- L'ensemble père est entièrement couvert par les ensembles fils
- Chaque noeud contient une "description" de son sous-arbre fournissant une **borne** inférieure pour $d(q, \cdot)$ dans le sous-ensemble correspondant



Branch and Bound: Recherche à un rayon près

Objectif: Trouver tous les p_i tels que $d(p_i, q) \leq r$:

1. Parcours en profondeur de l'arbre (algorithme récursif)
2. A chaque noeud, calcul de la borne inférieure du sous-ensemble correspondant
3. Elimination des branches avec une borne inférieure supérieure à r



B&B: Recherche du plus proche voisin

Objectif: trouver $\text{argmin}_{p_i}\ d(p_i, q)$:

1. Choisir aléatoirement un p_i, initialiser $p_{NN} := p_i, r_{NN} := d(p_i, q)$
2. Commencer une recherche à r_{NN} près
3. Lorsque l'on rencontre un p' tel que $d(p', q) < r_{NN}$, mise à jour $p_{NN} := p', r_{NN} := d(p', q)$

B&B: Best Bin First

Objectif: trouver $\text{argmin}_{p_i}\ d(p_i, q)$:

1. Choisir aléatoirement un p_i, initialiser $p_{NN} := p_i, r_{NN} := d(p_i, q)$
2. Mettre le noeud racine dans une **queue d'inspection**
3. Répéter: choisir le noeud de la queue d'inspection ayant la borne inférieure la plus petite, calculer les bornes inf. des sous-ensembles fils
4. Insérer dans le queue les fils dont la borne inf. est inférieure à r_{NN} ; éliminer les branches des autres fils
5. Lorsque l'on rencontre un p' tel que $d(p', q) < r_{NN}$, mise à jour $p_{NN} := p', r_{NN} := d(p', q)$

Structures basées sur des arbres !

Sphere Rectangle Tree k-d-B tree

Geometric near-neighbor access tree Excluded middle vantage point forest mvp-tree Fixed-height fixed-queries tree

Vantage-point tree

R*-tree Burkhard-Keller tree BBD tree Voronoi tree Balanced aspect ratio tree Metric tree vp^s-tree

M-tree

SS-tree R-tree Spatial approximation tree Multi-vantage point tree Bisector tree mb-tree

Generalized hyperplane tree

Hybrid tree Slim tree Spill Tree Fixed queries tree X-tree

k-d tree Balltree Quadtree Octree

SR-tree Post-office tree

Chapitre III

Techniques basiques pour les espaces Euclidiens

Avantages des espaces Euclidiens

- Formalisme mathématique riche pour définir les frontières des sous-ensembles d'objets

  Examples: rectangles, hyperplans, courbes polynomiales

- Calcul facilité de la borne inférieure de la distance entre la requête et les frontières d'un ensemble

- Facilite l'utilisation de projections dans des espaces réduits

k-d Tree

Construction: Bentley, 1975

Partitionnement Top-down (objets insérés un par un)
A chaque niveau l: partitionne l'ensemble courant par un hyperplan orthogonal à l'axe $l \bmod k$

k-d Tree

Construction: Bentley, 1975

Partitionnement Top-down (objets insérés un par un)
A chaque niveau l: partitionne l'ensemble courant par un hyperplan orthogonal à l'axe $l \bmod k$

Traitement Requête:

Branch and bound standard

k-d Tree

Construction: Bentley, 1975

Partitionnement Top-down (objets insérés un par un)
A chaque niveau l: partitionne l'ensemble courant par un hyperplan orthogonal à l'axe $l \bmod k$

Traitement Requête:

Branch and bound standard

k-d Tree

Construction: Bentley, 1975

Partitionnement Top-down (objets insérés un par un)
A chaque niveau l: partitionne l'ensemble courant par un hyperplan orthogonal à l'axe $l \bmod k$

Traitement Requête:

Branch and bound standard

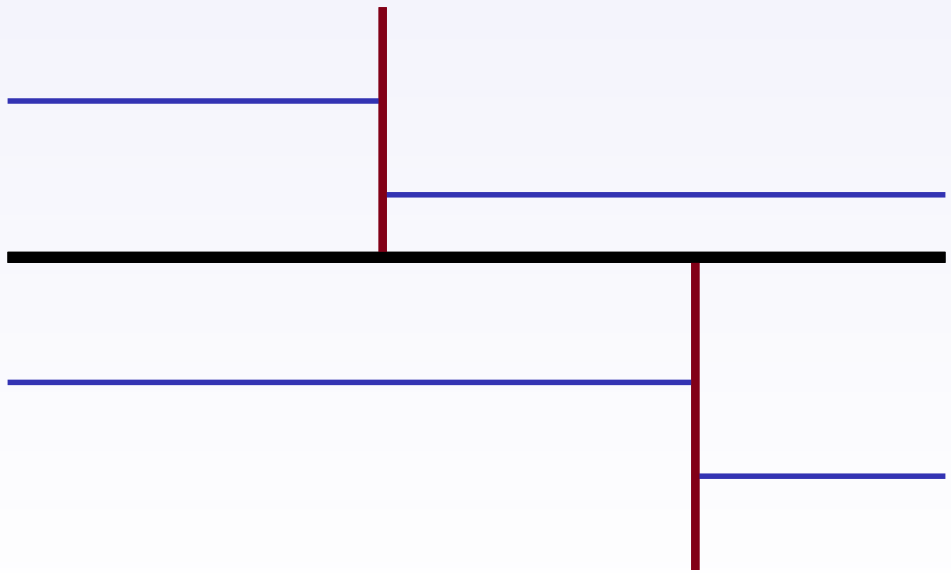
k-d Tree

Construction: Bentley, 1975

Partitionnement Top-down (objets insérés un par un)
A chaque niveau l: partitionne l'ensemble courant par un hyperplan orthogonal à l'axe $l \bmod k$

Traitement Requête:

Branch and bound standard



R-Tree

Construction: Guttman, 1984

Partitionnement Top-down ou Bottom-up
Rectangles englobant minimum de groupements d'objets
A chaque niveau: regroupe les rectangles proches
et calcule le rectangle minimum englobant

R-Tree

Construction: Guttman, 1984

Partitionnement Top-down ou Bottom-up
Rectangles englobant minimum de groupements d'objets
A chaque niveau: regroupe les rectangles proches
et calcule le rectangle minimum englobant

Query processing:

Branch and bound standard

R-Tree

Construction: Guttman, 1984

Partitionnement Top-down ou Bottom-up
Rectangles englobant minimum de groupements d'objets
A chaque niveau: regroupe les rectangles proches
et calcule le rectangle minimum englobant

Query processing:

Branch and bound standard

Insertions/Suppressions: similaire à un B-tree

R-Tree

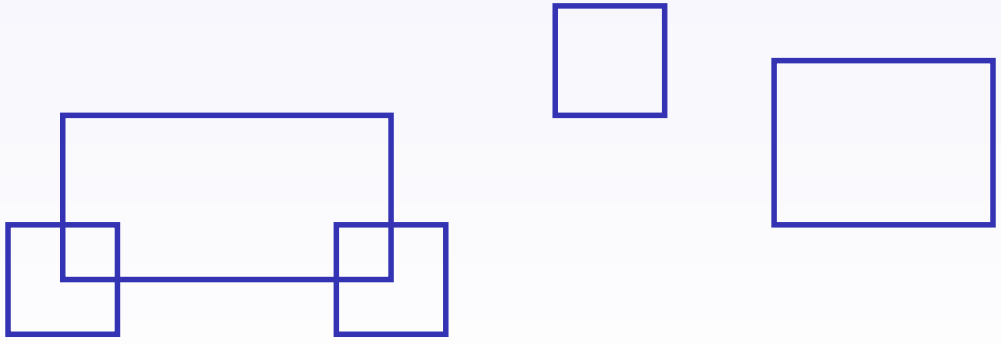
Construction: Guttman, 1984

Partitionnement Top-down ou Bottom-up
Rectangles englobant minimum de groupements d'objets
A chaque niveau: regroupe les rectangles proches
et calcule le rectangle minimum englobant

Query processing:

Branch and bound standard

Insertions/Suppressions: similaire à un B-tree



R-Tree

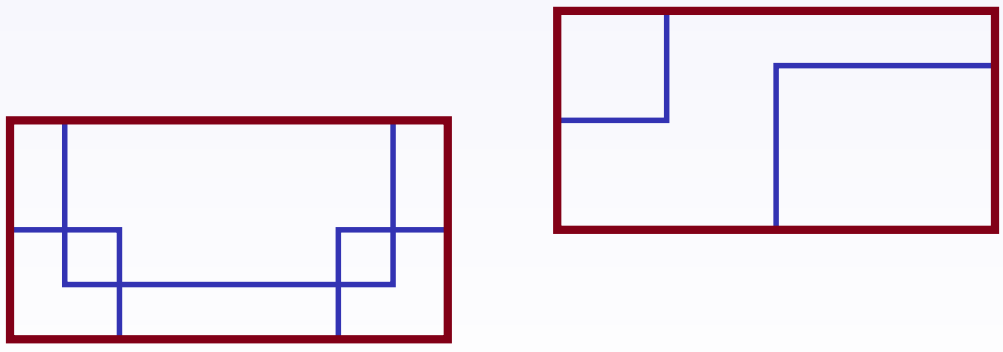
Construction: Guttman, 1984

Partitionnement Top-down ou Bottom-up
Rectangles englobant minimum de groupements d'objets
A chaque niveau: regroupe les rectangles proches
et calcule le rectangle minimum englobant

Query processing:

Branch and bound standard

Insertions/Suppressions: similaire à un B-tree



R-Tree

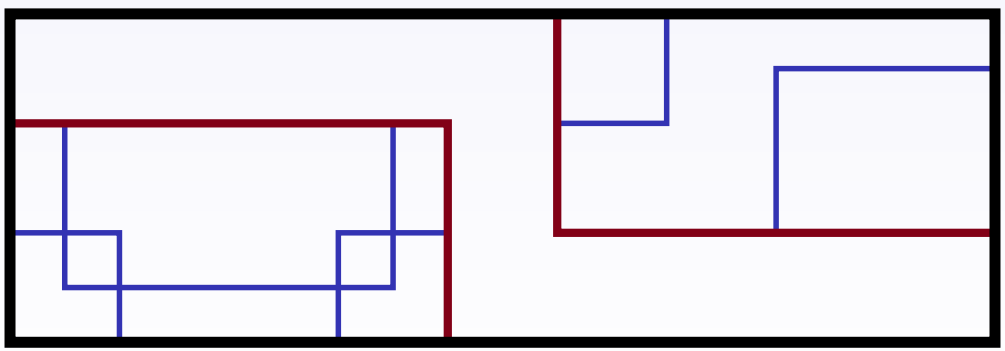
Construction: Guttman, 1984

Partitionnement Top-down ou Bottom-up
Rectangles englobant minimum de groupements d'objets
A chaque niveau: regroupe les rectangles proches
et calcule le rectangle minimum englobant

Query processing:

Branch and bound standard

Insertions/Suppressions: similaire à un B-tree



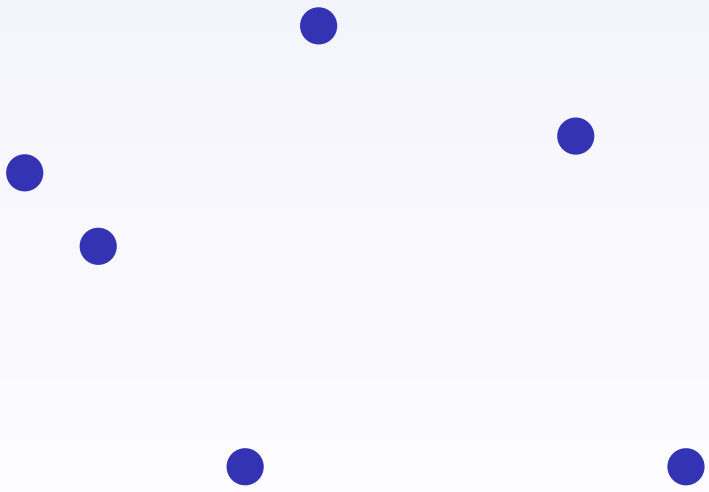
Chapitre IV

Arbres basés sur des pivots (Vantage-Point Trees et dérivés)

Technique de partition

Uhlmann'91, Yianilos'93:

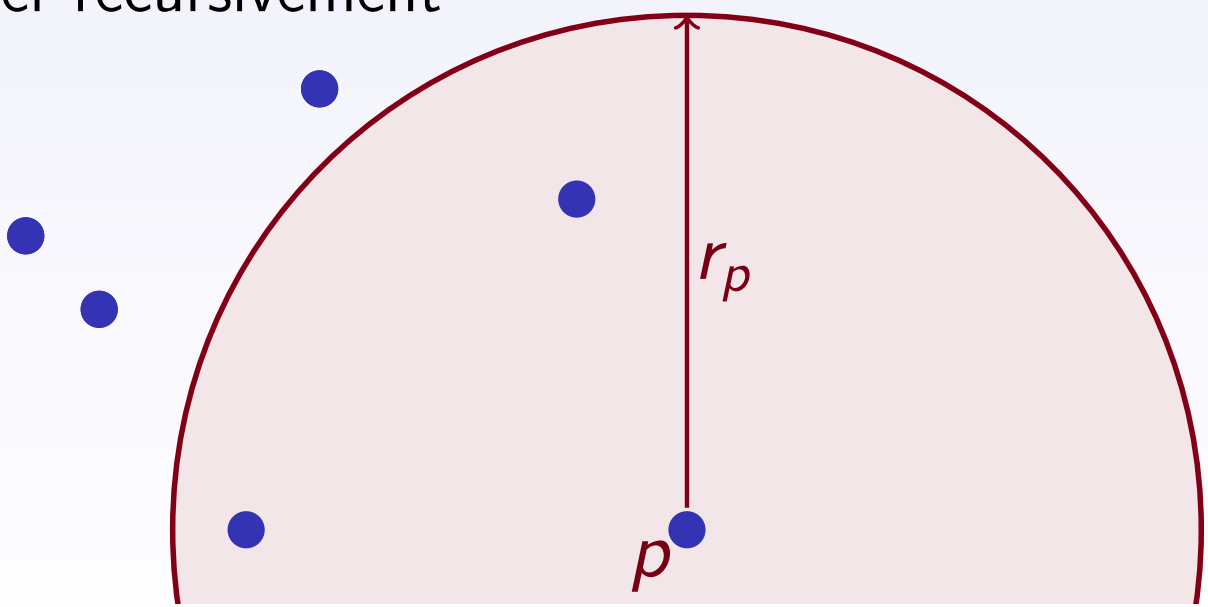
1. Choisir un objet p de la base (appelé pivot)
2. Choisir un rayon de partition r_p
3. Mettre les p_i tels que $d(p_i, p) \leq r$ dans une branche ("intérieure"), les autres dans une autre branche ("extérieure")
4. Répéter récursivement



Technique de partition

Uhlmann'91, Yianilos'93:

1. Choisir un objet p de la base (appelé pivot)
2. Choisir un rayon de partition r_p
3. Mettre les p_i tels que $d(p_i, p) \leq r$ dans une branche ("intérieure"), les autres dans une autre branche ("extérieure")
4. Répéter récursivement



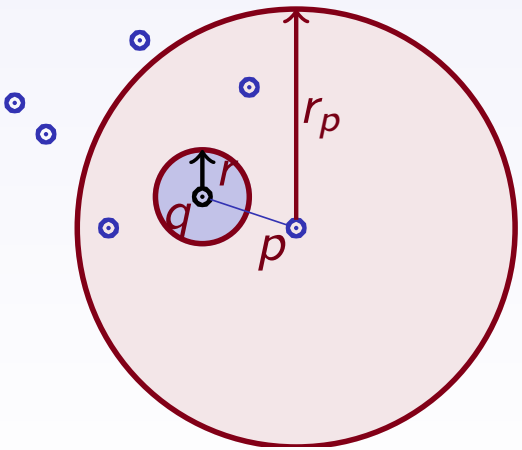
Condition de filtrage des branches

Pour une requête à r-près:

Si $d(q,p) > r_p + r$ élimine la branche intérieure

Si $d(q,p) < r_p - r$ élimine la branche extérieure

Pour $r_p - r \leq d(q,p) \leq r_p + r$ il faut inspecter les deux branches



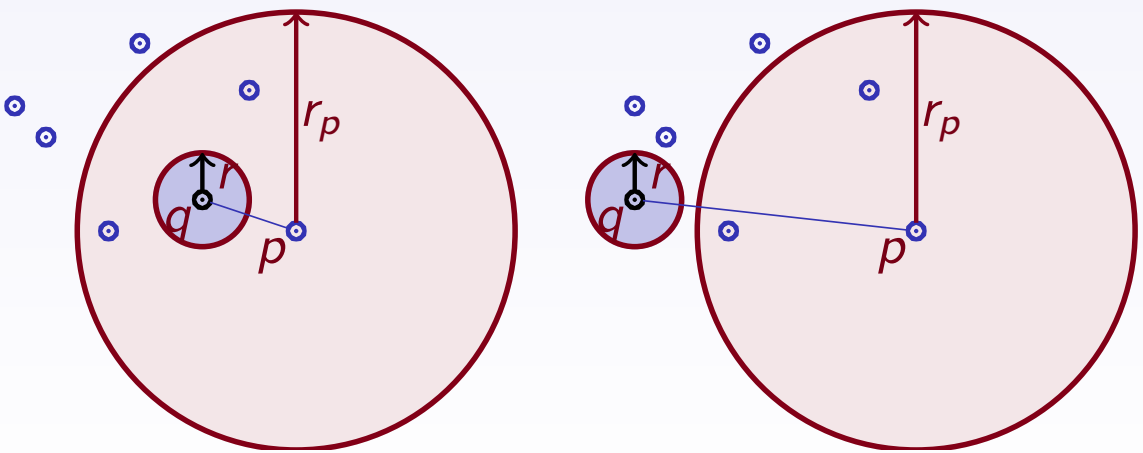
Condition de filtrage des branches

Pour une requête à r-près:

Si $d(q,p) > r_p + r$ élimine la branche intérieure

Si $d(q,p) < r_p - r$ élimine la branche extérieure

Pour $r_p - r \leq d(q,p) \leq r_p + r$ il faut inspecter les deux branches



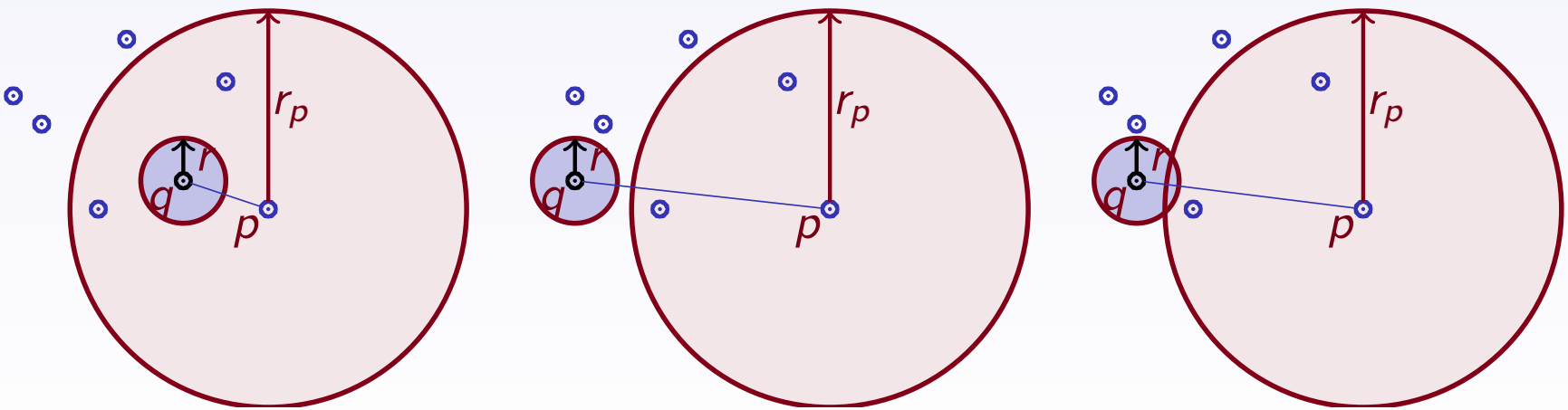
Condition de filtrage des branches

Pour une requête à r-près:

Si $d(q,p) > r_p + r$ élimine la branche intérieure

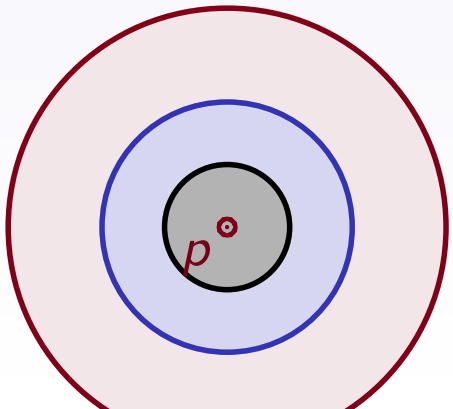
Si $d(q,p) < r_p - r$ élimine la branche extérieure

Pour $r_p - r \leq d(q,p) \leq r_p + r$ il faut inspecter les deux branches



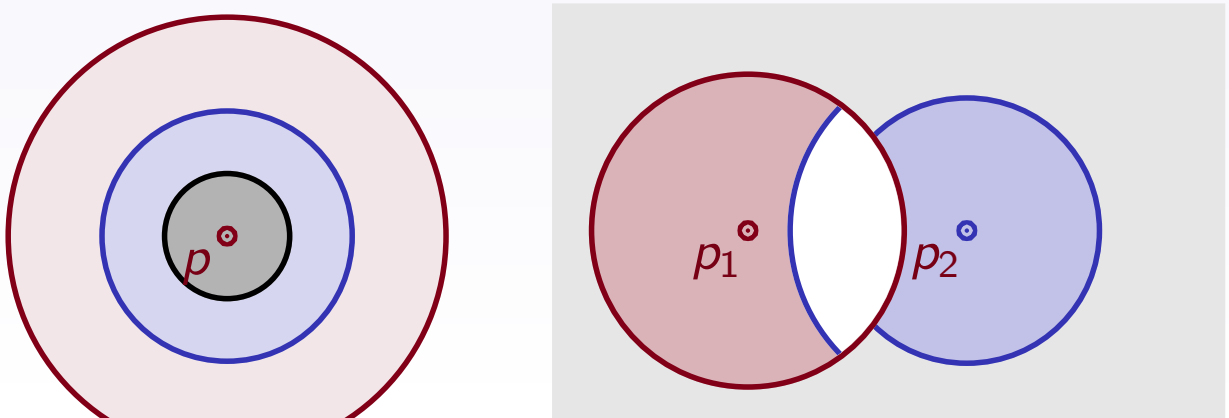
Variations du "Vantage-Point Tree"

- **Burkhard-Keller tree:** pivot utilisé pour divisé l'espace en m anneaux Burkhard&Keller'73
- **MVP-tree:** utilisation du même pivot pour différents noeuds du même niveau
  Bozkaya&Ozsoyoglu'97
- **Post-office tree:** utilise $r_p + \delta$ pour la branche intérieure, $r_p - \delta$ pour la branche extérieure
  McNutt'72



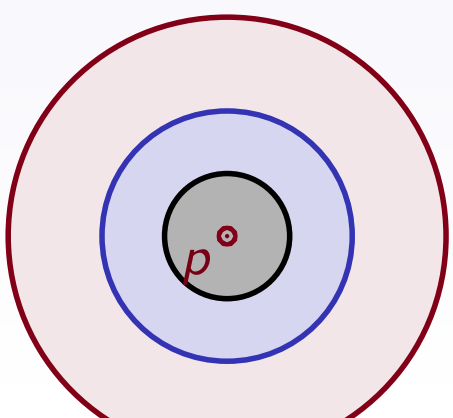
Variations du "Vantage-Point Tree"

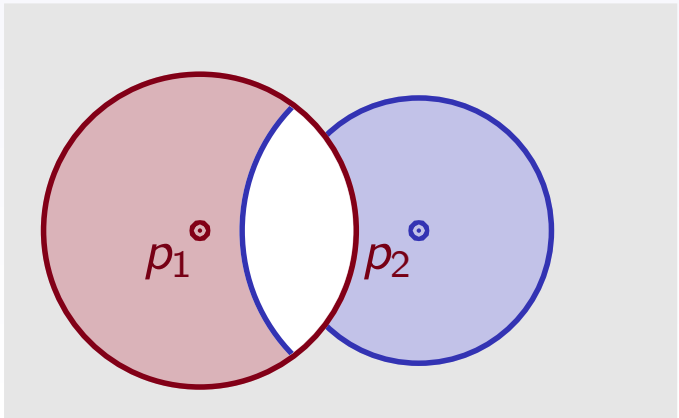
- **Burkhard-Keller tree:** pivot utilisé pour divisé l'espace en m anneaux Burkhard&Keller'73
- **MVP-tree:** utilisation du même pivot pour différents noeuds du même niveau
  Bozkaya&Ozsoyoglu'97
- **Post-office tree:** utilise $r_p + \delta$ pour la branche intérieure, $r_p - \delta$ pour la branche extérieure
  McNutt'72

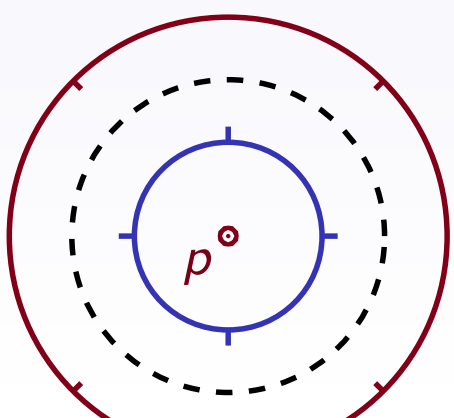


Variations du "Vantage-Point Tree"

- **Burkhard-Keller tree:** pivot utilisé pour divisé l'espace en m anneaux Burkhard&Keller'73
- **MVP-tree:** utilisation du même pivot pour différents noeuds du même niveau
  Bozkaya&Ozsoyoglu'97
- **Post-office tree:** utilise $r_p + \delta$ pour la branche intérieure, $r_p - \delta$ pour la branche extérieure
  McNutt'72







Chapter V

Arbres Hyperplan généralisés et dérivés

Arbre Hyperplan généralisé

Technique de partition (Uhlmann'91):

- Choisir deux pivots p_1 and p_2
- Mettre tous les objets plus proches de p_1 que de p_2 dans la branche de gauche, et les autres dans celle de droite
- Répéter récursivement

Arbre Hyperplan généralisé

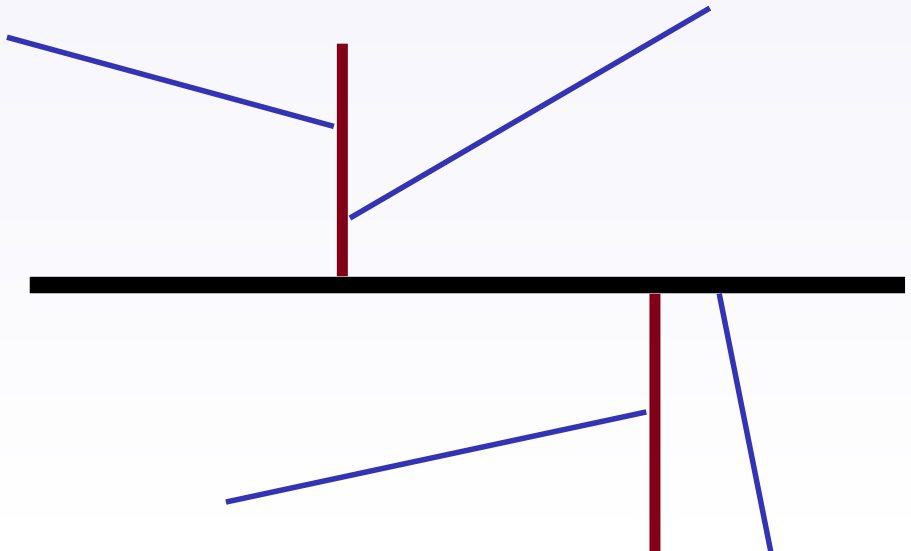
Technique de partition (Uhlmann'91):

- Choisir deux pivots p_1 and p_2
- Mettre tous les objets plus proches de p_1 que de p_2 dans la branche de gauche, et les autres dans celle de droite
- Répéter récursivement

Arbre Hyperplan généralisé

Technique de partition (Uhlmann'91):

- Choisir deux pivots p_1 and p_2
- Mettre tous les objets plus proches de p_1 que de p_2 dans la branche de gauche, et les autres dans celle de droite
- Répéter récursivement



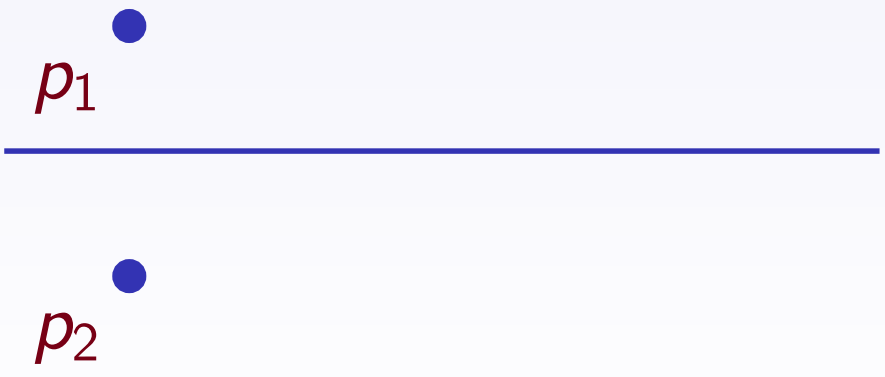
GH-Tree: Condition de filtrage des branches

Pour une requête à r-près:

Si $d(q, p_1) > d(q, p_2) + 2r$ élimine la branche gauche

Si $d(q, p_1) < d(q, p_2) - 2r$ élimine la branche droite

Pour $|d(q, p_1) - d(q, p_2)| \leq 2r$ il faut inspecter les deux branches



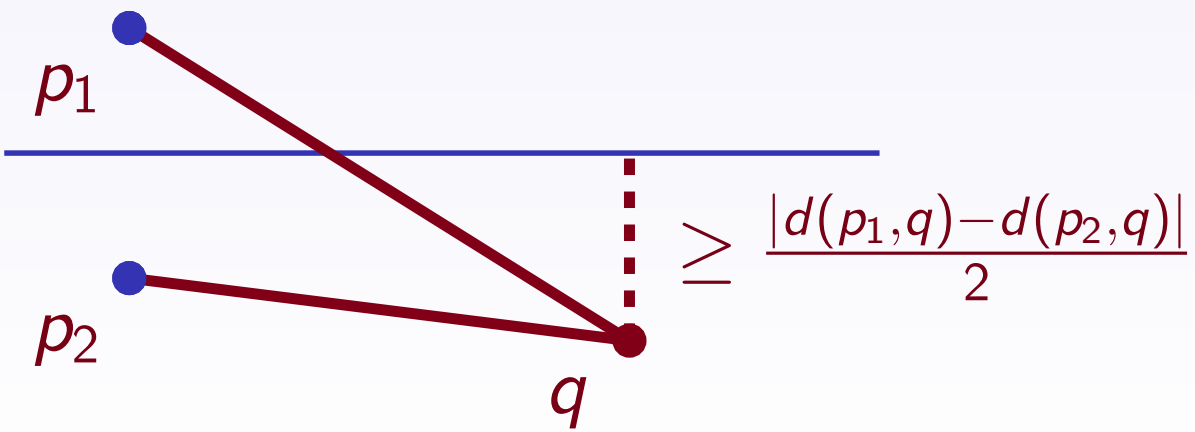
GH-Tree: Condition de filtrage des branches

Pour une requête à r-près:

Si $d(q, p_1) > d(q, p_2) + 2r$ élimine la branche gauche

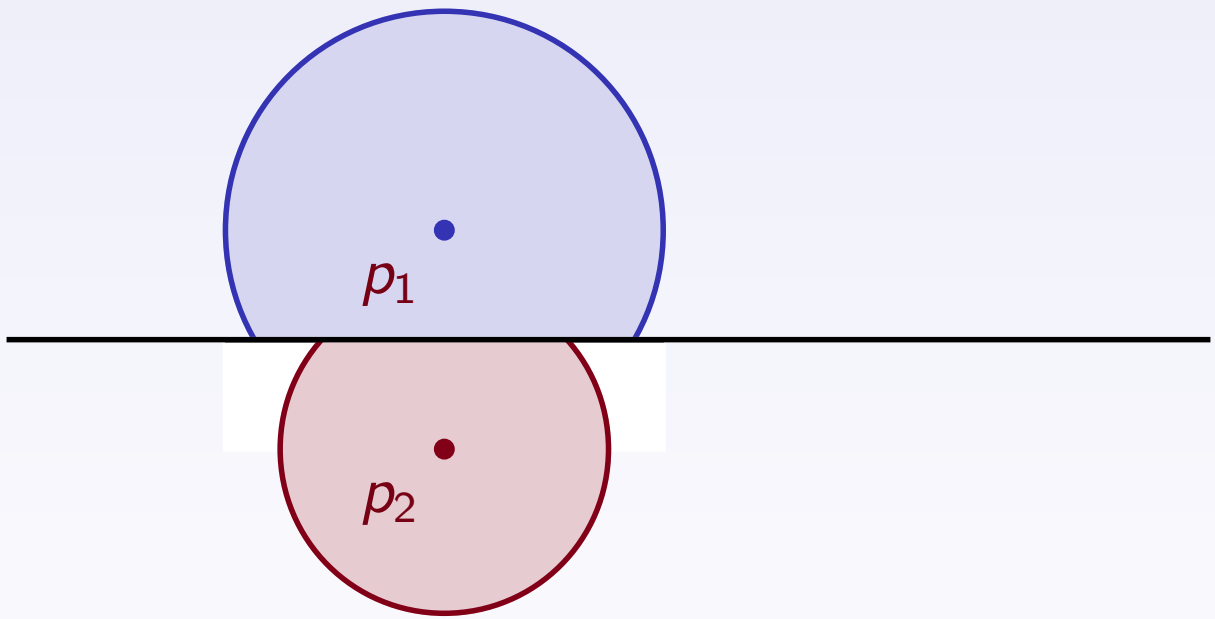
Si $d(q, p_1) < d(q, p_2) - 2r$ élimine la branche droite

Pour $|d(q, p_1) - d(q, p_2)| \leq 2r$ il faut inspecter les deux branches



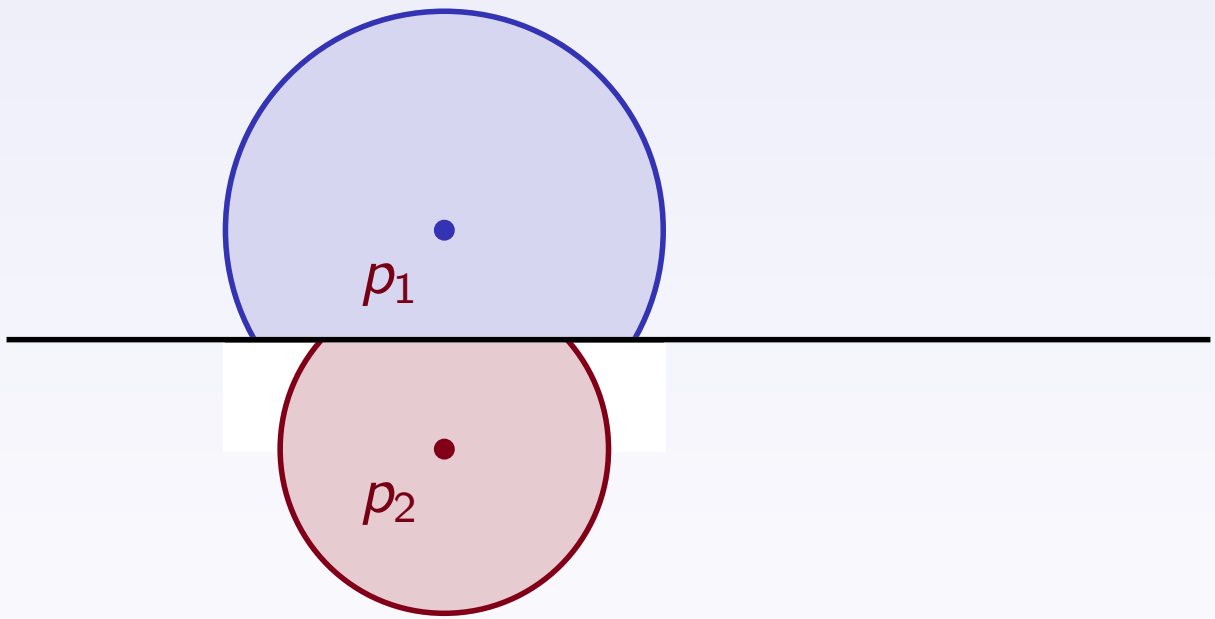
Bisector trees

Si l'on conserve les rayons de couverture pour p_1 et la branche de gauche, et pour p_2 et la branche de droite: Conditions de filtrage plus efficace



Bisector trees

Si l'on conserve les rayons de couverture pour p_1 et la branche de gauche, et pour p_2 et la branche de droite: Conditions de filtrage plus efficace



Variation: bisector tree monotone (Noltemeier, Verbarg, Zirkelbach'92) utilise toujours le pivot parent comme un des deux fils

Exercices

Exercices

Prouver que la borne inférieure d'un GH-tree est valide

Exercices

Prouver que la borne inférieure d'un GH-tree est valide

Prouver que le rayon de couverture est décroissant dans un bisector tree monotone

Exercices

Prouver que la borne inférieure d'un GH-tree est valide

Prouver que le rayon de couverture est décroissant dans un bisector tree monotone

Construire une base et un ensemble de requêtes dans l'espace Euclidien pour lesquels toutes les structures décrites requièrent un temps de recherche du plus proche voisin en $\Omega(n)$

En bref

- La recherche par similarité est fondamentale pour la recherche d'information, la fouille de données, l'apprentissage et les systèmes de recommendation

En bref

- La recherche par similarité est fondamentale pour la recherche d'information, la fouille de données, l'apprentissage et les systèmes de recommendation
- Boules, rectangles, hyperplans (généralisés) sont utilisés pour le partitionnement des données sous forme d'arbres

En bref

- La recherche par similarité est fondamentale pour la recherche d'information, la fouille de données, l'apprentissage et les systèmes de recommendation
- Boules, rectangles, hyperplans (généralisés) sont utilisés pour le partitionnement des données sous forme d'arbres
- Parcours en profondeur et stratégie best-bin-first sont utilisées pour la recherche

En bref

- La recherche par similarité est fondamentale pour la recherche d'information, la fouille de données, l'apprentissage et les systèmes de recommendation
- Boules, rectangles, hyperplans (généralisés) sont utilisés pour le partitionnement des données sous forme d'arbres
- Parcours en profondeur et stratégie best-bin-first sont utilisées pour la recherche

Merci de votre attention! Questions?

Références

The Homepage of Nearest Neighbors and Similarity Search
http://simsearch.yury.name

P. Zezula, G. Amato, V. Dohnal, M. Batko
Similarity Search: The Metric Space Approach. Springer, 2006.
http://www.nmis.isti.cnr.it/amato/similarity-search-book/

E. Chávez, G. Navarro, R. Baeza-Yates, J. L. Marroquín
Searching in Metric Spaces. ACM Computing Surveys, 2001.
http://www.cs.ust.hk/~leichen/courses/comp630j/readings/acm-survey/searchinmetric.pdf

G.R. Hjaltason, H. Samet
Index-driven similarity search in metric spaces. ACM Transactions on Database Systems, 2003
http://www.cs.utexas.edu/~abhinay/ee382v/Project/Papers/ft_gateway.cfm.pdf