계층적 군집화를 이용한 근사 단어 필터링 기법

김성환, 조환규
부산대학교 컴퓨터공학과
e-mail:{sunghwan,hgcho}@pusan.ac.kr

Proximate Word Filtering by Hierarchical Clustering

Sung-Hwan Kim, Hwan-Gue Cho
Pusan National University

요 약

단어 필터링은 유해정보를 차단위한 기본적인 기능이다. 그러나 악의적인 사용자는 필터링 시스템을 우회하기 위하여 금지 단어에 의도적인 변형을 가한다. 이에 대응하기 위해 일정 오류를 허용하여 필터링을 수행하는 근사 단어 필터링이 있다. 근사 단어를 검색하기 위한 문자열 색인 방법으로는 주로 기준 단어(Pivot)을 이용한 유클리드 공간에의 사상을 이용하는데, 이는 단어 필터링에 응용하기에는 근본적인 구조상의 한계점이 있다. 본 논문에서는 필터링 대상이 되는 단어 집합 내에서 군집화를 수행하여 계층적인 자료구조를 구성하고, 단어 필터링을 위한 필터링 질의(Filtering query)를 정의한 뒤 그에 적합한 탐색 상의 적용에 관하여 설명한다. 실험 결과 기존의 기준 단어(Pivot)을 이용한 색인 기법에 비하여 16.9%~26.6%의 탐색 속도 향상을 확인할 수 있었다.

1. 문제 정의

음란물, 스팸 메일, 욕설 등과 같은 불건전한 정보를 차단하기 위한 가장 기본적인 방법은 단어 필터링을 이용하는 것이다. 유해정보들에서만 빈번하게 등장하는 단어를 미리 금지 단어 목록에 등록한 후, 금지 단어 목록에 존재하는 단어가 출현하면 해당 정보를 차단한다.

완전 일치(Exact Matching)를 기반으로 하는 문자열 비교 알고리즘이나 정규식 등을 이용하는 경우 단어 필터링 기능에 취약점이 발생한다[1]. 악의적인 사용자는 금지 단어 목록에 등록된 단어를 교묘하게 변형시킬 수 있다. 특히 한글의 경우에는 유사 음소 간의 변형을 통해 의미는 유지하면서도 형태만 미묘하게 다른 유사 단어를 이용하여 필터링 시스템을 우회하기 쉽다[2].

이러한 문제에 대응하기 위하여 근사 단어 필터링 기법이 제안되어 왔다[1,3]. 근사 단어 필터링은 등록된 금지 단어 집합 내의 단어 중 주어진 질의 단어와의 거리가 일정 반경 이하인 단어가 존재하는 지에 대한 문제로, 모든 문자열의 집합을 Σ^*라 할 때, 다음과 같이 형식적으로 정의할 수 있다.

문제 Approximate Word Filtering
입력 $S \subset \Sigma^*$: 금지 문자열 집합
$q \in \Sigma^*$: 질의 문자열
d : 문자열 간 거리 함수 $d: \Sigma^* \times \Sigma^* \rightarrow \mathbb{R}$
$r \in \mathbb{R}$: 기준 반경
출력 *True,* if $\exists s \in S$ s.t. $d(s,q) \leqq r$
False, otherwise

2. 거리 공간(Metric Space)과 문자열 편집 거리

어떤 집합 S 상에서 정의된 거리 함수 $d: S \times S \rightarrow R$가 있을 때, d가 다음 조건을 만족하는 경우 $\langle S,d \rangle$를 거리공간(Metric Space)이라 한다.

1. $d(x,y) = 0$ iff $x = y$
2. $d(x,y) = d(y,x)$
3. $d(x,y) + d(y,z) \geqq d(x,z)$

다양한 탐색 기법들이 위 성질을 만족하는 거리 함수를 전제로 하고 있다. 특히 3번의 삼각부등식 성질을 이용하면 탐색 공간을 축소시킬 수 있기 때문에 주어진 공간 질의에 대한 전체적인 탐색 속도를 향상시키는데 유리하다.

문자열 편집 거리(Edit distance, Levenshtein distance)는 거리 공간 성질을 만족하는 대표적인 거리 함수이다. 두 문자열의 편집 거리는 어느 한 문자열로부터 다른 문자열로 만들기 위해 필요한 최소 연산(삽입, 삭제, 치환) 횟수로 다음과 같은 재귀식에 따른 동적계획법(Dynamic programming)에 의하여 계산된다.

$$EditDistance(\boldsymbol{x},\boldsymbol{y}) = Ed(|\boldsymbol{x}|,|\boldsymbol{y}|;\ \boldsymbol{x},\boldsymbol{y})$$
$$Ed(i,0;\ \boldsymbol{x},\boldsymbol{y}) = i$$
$$Ed(0,j;\ \boldsymbol{x},\boldsymbol{y}) = j$$
$$Ed(i+1,j+1;\ \boldsymbol{x},\boldsymbol{y}) = \min\begin{Bmatrix} Ed(i,j+1;\ \boldsymbol{x},\boldsymbol{y})+1 \\ Ed(i+1,j;\ \boldsymbol{x},\boldsymbol{y})+1 \\ Ed(i,j;\ \boldsymbol{x},\boldsymbol{y})+\sigma(\boldsymbol{x}[i],\boldsymbol{y}[j]) \end{Bmatrix}$$

단, $\sigma(\alpha,\beta) = \begin{cases} 0, \alpha = \beta \\ 1, \alpha \neq \beta \end{cases}$

3. 관련 연구: 기준 단어를 이용한 근사 탐색 기법

문자열 간에 거리 공간(Metric space)의 성질을 만족하는 거리 함수가 주어지면 이미 잘 알려진 거리 공간 상에서의 탐색 기법을 이용하여 질의를 수행할 수 있다[4].

3.1 다차원 공간으로의 사상[5]: Kd-Tree, R-Tree

k개의 기준 단어(Pivot) $p_1,\cdots,p_k$을 설정하면 단어 w에 대하여 k차원 공간 좌표 $\langle d(w,p_1),d(w,p_2),\cdots,d(w,p_k)\rangle$에 대응시킬 수 있다. 이 때, 주어진 질의 단어 q에 대하여 $d(q,x)\leq r$인 x가 존재하는지를 찾기 위해서는 대응되는 공간상에서 $\langle d(q,p_1)\pm r,\cdots,d(q,p_k)\pm r\rangle$의 범위 내에 위치한 x에 대해서만 $d(q,x)$를 확인해보면 된다. 해당 범위를 벗어나는 x는 삼각부등식 성질에 의하여 절대 $d(q,x)\leq r$을 만족할 수 없다. Kd-Tree나 R-Tree를 이용하여 사상된 공간 좌표에 대한 탐색을 수행할 수 있다.

3.2 거리 기반의 분할[6]: Vp-Tree

단어 집합 S가 있을 때, S의 원소 중 어떤 기준 단어 p를 중심으로 적당한 반경 ρ내에 있는 단어들의 집합 $S_{IN}=\{x\in S: d(x,p)\leq\rho\}$과 반경 밖에 있는 단어들의 집합 $S_{OUT}=\{x\in S: d(x,p)>\rho\}$으로 분할할 수 있다. 이 때, 질의 단어 q가 주어졌을 때, 만약 $d(q,p)>r+\rho$이면 삼각부등식에 의해 S_{IN}내에는 $d(q,x)\leq r$를 만족하는 x가 존재하지 않으며, 반대로 $d(q,p)\leq r-\rho$이면, S_{OUT}내에 해당하는 x가 존재하지 않음을 알 수 있다. Vp-Tree를 이용하면 분할된 집합 S_{IN}과 S_{OUT}에 대하여도 각각 반복적으로 트리 형태의 구조를 구성하여 탐색을 수행할 수 있다.

3.3 기존 방법의 한계

기존의 유클리드 다차원 공간으로의 사상이나 거리 기반의 분할을 이용한 탐색 기법의 경우에는 어떤 기준 단어(Pivot)을 선택하느냐에 따라 성능의 차이가 발생한다. 일반적인 기준 선택 방법에 대해서는 현재에도 계속 연구가 진행 중에 있다. 더욱이 효율적인 기준점으로 이상점(Outlier)을 이용하는 것이 일반적인데[7], 이는 변형 단어가 다수 존재하는 필터링 데이터베이스 상에서의 탐색에 적합하지 않을 수 있다.

기존의 기법들이 주로 다루는 최근접이웃탐색(Nearest neighbor search)이나 범위 탐색(Range search) 문제는 해당 조건을 만족하는 정확한 원소 또는 원소의 집합을 구하는 것이 목적이다. 그에 비하여 본 논문에서 다루고자 하는 문제는 일정 반경 내에 원소가 존재하는지에 대한 여부만을 알아내는 것이 목적이다. 기본적으로는 기존 기법을 이용하여 질의 단어 q의 최근접이웃 x^*를 찾은 후 두 단어 간의 거리 $d(q,x^*)$가 기준 반경 r이하인지를 살펴보면 되지만 최근접이웃을 정확히 구하기 위한 불필요한 연산이 추가로 발생하게 된다는 문제점이 있다.

4. 제안 기법: 단어 집합의 계층적 군집화

본 장에서는 먼저 계층적인 군집화 기법을 이용한 트리(Tree)형태의 자료구조의 상향식 구성 방법을 제안하고 다음 장에서 탐색 방법에 대하여 설명하고자 한다.

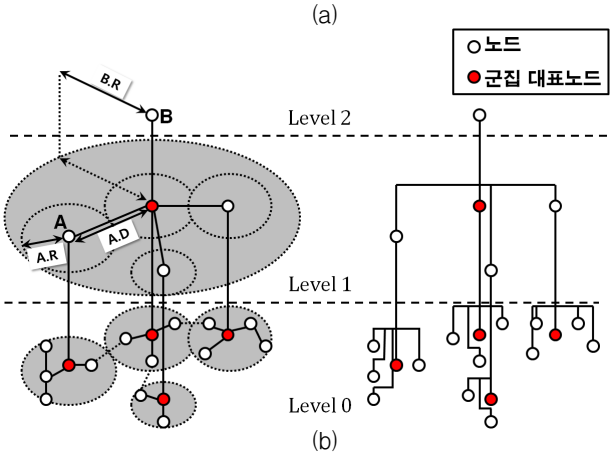
먼저 트리의 각 노드들이 가지는 속성은 다음과 같다. 하위 노드들의 집합인 Children과 자신의 하위 노드들 가운데 가장 멀리 떨어진 노드를 포함하기 위한 반경 R, 자신의 부모노드와의 거리 D를 속성으로 가진다.

전반적인 구축 알고리즘은 그림 1과 같다. 먼저 주어진 단어 집합의 각 단어들은 트리의 말단 노드(Leaf node)가 된 후 집합 W에 포함된다. 이후에는 집합 W에 있는 노드들에 대하여 군집화를 수행한 뒤 각 군집 C_i에 대하여 대표 단어를 선발하고 군집 내 모든 단어 노드들에 대하여 군집 대표 단어와의 거리 D를 계산한다. 각 군집의 대표 단어는 자기 자신을 포함하여 군집 내 모든 단어 노드

Algorithm ConstructHCTree
Input S : 단어 집합
$d()$: 단어 간 거리 함수
$cluster()$: 군집화 함수
$center()$: 군집 대표 추출 함수
Output t : 트리의 root 노드

$W\leftarrow\phi$
For each $s\in S$
  $n\leftarrow$**new** Node(s)
  n.R$\leftarrow 0$
  $W\leftarrow W\cup\{n\}$
while $|W|>1$
  $W^*\leftarrow\phi$
  $\{C_i\}\leftarrow cluster(W)$
  For each $C_i\in\{C_i\}$
    $c^*\leftarrow center(C_i)$
    $n^*\leftarrow$**new** Node(c^*.data)
    For each $n\in C_i$
      n.D$\leftarrow d(n, n^*)$
      n^*.Children.add(n)
    n^*.R$\leftarrow \max_{n\in C_i}(n.D+n.R)$
    $W^*\leftarrow W^*\cup\{n^*\}$
  $W\leftarrow W^*$
$t\leftarrow w$, where $W=\{w\}$

(a)



(b)

(그림 1) 계층적 군집화를 이용한 트리 구성 방법. (a)트리 구성 알고리즘과 (b)트리 구성 개념도. 가장 하위 레벨에서부터 군집화를 수행한 후 각 군집으로부터 대표 단어를 추출하여 다음 단계로 진행한다.

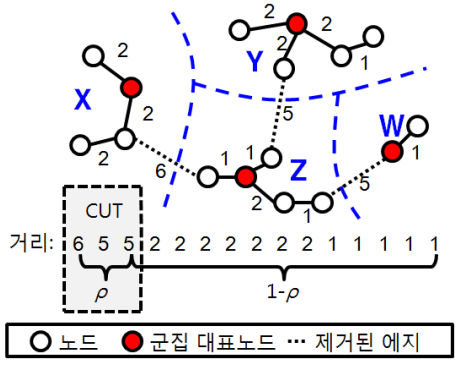
들을 하위 노드로 가지게 되며, 각 하위 노드를 중심으로 한 반경을 모두 포함할 수 있는 최소한의 반경을 자기 자신의 반경 R로 가진다. 다음 단계에서는 이전 단계의 군집 대표 단어들의 집합을 W로 하여 같은 과정을 반복하여, 집합 W에 유일한 원소가 남을 때까지 상향식으로 트리를 구성한다.

트리 구축 과정에서 사용되는 군집화 함수 cluster()는 N개의 단어 노드를 가지는 집합 W에 대하여 $M(<N)$개의 군집으로 분할하며, $\cap_{i=0}^{m-1} C_i = \phi$, $\cup_{i=0}^{m-1} C_i = W$인 각 군집 $C_i \subset W$에 대하여 $\{C_i\}$의 형태의 출력을 가진다. 한편 각 군집으로부터 대표 단어 노드를 선출하는 함수 center()는 주어진 군집 C에 대하여 하나의 원소 $c^* \in C$를 주어진 조건에 맞게 선출하는 함수이다.

5. 군집화 및 대표 노드 선택 방법

4장에서 언급한 두 함수 cluster()와 center()에 관한 설명은 해당 함수들이 만족해야 할 최소한의 조건이다. 군집화, 대표노드 선출을 하는 구체적인 조건은 필요에 따라 변형하여 사용하면 된다.

본 논문에서 사용할 군집화 및 대표 노드 선택 방법은 그림 2에 나타나있다. 군집화 방법은 다음과 같다. 먼저 각 노드 쌍 $\{n_1,n_2\}$에 대하여 에지(Edge) $e=\{n_1,n_2\}$를 구성하고, 해당 에지에 대한 비용을 $\mathrm{cost}(e)=d(n_1,n_2)$라 한다. 이와 같이 구성된 그래프에 대하여 최소신장트리(Minimum Spanning Tree)를 $T=\langle V,E\rangle$를 구하고, $\mathrm{cost}(e)$값이 상위 ρ의 비율에 해당되는 에지들을 모두 제거함으로써 군집화를 수행한다. 즉, 주어진 실수 $\rho \in [0,1)$에 대하여 상위 $\rho|E|$개의 에지들을 제거한 후 각각의 서로 연결된 부분(Connected component)을 하나의 군집으로 설정한다. 그림 2에서는 최소신장트리가 구성된 이후 $\rho=0.2$로 군집화를 수행한 결과이다. 0.2×15=3이므로, 거리값 $\mathrm{cost}(e)$가 가장 높은 상위 3개의 에지들을 제거함으로써 4개의 군집 W,X,Y,Z를 얻을 수 있다.



(그림 2) 최소신장트리(Minimum Spanning Tree)로부터 군집화를 수행하고 대표노드를 선택하는 방법. 에지에 할당된 거리값들을 크기순으로 정렬한 후 거리값이 높은 에지를 ρ의 비율만큼 제거한다.

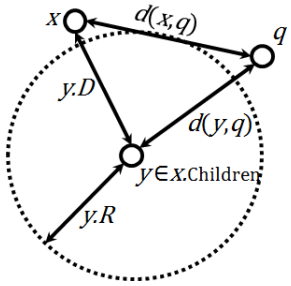
각 군집의 대표 노드는 해당 군집 내의 모든 노드들과의 거리의 합이 최소가 되는 노드를 선택한다. 다시 말해, 어떤 군집 $C=\{n_1,\cdots,n_M\}$이 있을 때, 군집 C의 대표 노드 $center(C)$는 다음과 같다.

$$center(C)=\arg\min_{m\in C}\sum_{n\in C} d(m,n)$$

만약 조건을 만족하는 노드가 하나 이상 존재하는 경우에는 그 중 하나를 임의로 선택한다.

6. 최근접이웃 탐색 과정을 이용한 필터링 기법

주어진 질의어 q와 필터링 반경 r에 대하여 $d(s,q)\le r$인 $s\in S$가 존재하는지에 대한 여부를 탐색하는 과정은 최상우선 탐색에 기반하며, 자세한 내용은 다음과 같다. 먼저 루트 노드(Root node)를 우선순위 대기열에 삽입한다. 이후 우선순위 대기열로부터 가장 작은 키(Key)를 가지는 노드 n_{top}을 찾는다. 만약 $d(n_{\mathrm{top}},q)\le r$이라면 탐색은 중단된다. 그렇지 않은 경우 해당 노드가 가지는 R(모든 하위 노드들을 포함하는 반경)과 주어진 필터링 반경 r의 합보다 작은 경우에는 하위 노드 중 하나 이상이 필터링 반경 내에 포함되어 있을 가능성이 있으므로 모든 자식 노드 x를 $|d(n_{\mathrm{top}},q)-x.D|$의 키 값과 함께 우선순위 대기열에 추가한다. 그림 3과 같이 삼각부등식의 성질에 의하여 n_{top}의 자식노드 x의 키 값은 항상 $d(x,q)$보다 작기 때문에 우선순위 대기열에서 가장 작은 키 값을 가진 노드가 말단 노드인 경우 해당 노드가 최근접이웃임은 자명하다. 필터링 과정에 대한 알고리즘은 그림 4에 기술되어 있다.



(그림 3) 삼각부등식 성질을 이용한 하위 노드와의 거리 추정. $d(y,q)$는 항상 $|d(x,q)-y.D|$보다 크거나 같다.

Algorithm RangeFiltering
Input q : 질의 단어
r : 필터링 반경
T : 계층적 군집화 트리
Output $True/False$: $d(q,x)\le r$인 x의 존재 여부

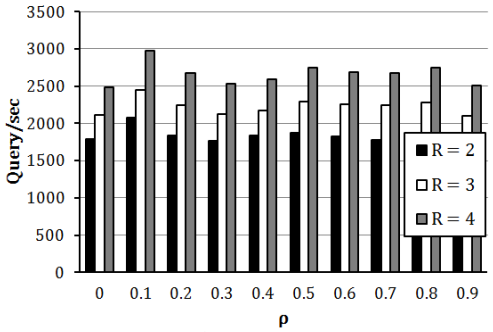
```
Q←new PriorityQueue()
Q.push( T.root with Key=0 )
while not Q.empty()
    t←Q.pop() // pop the node with the lowest key
    if d(q,t) ≤ r then return true
    if d(q,t) ≤ t.R + r then
        for each n∈t.Children
            Q.push( n with Key=abs(d(q,t)-n.D))
return false
```

(그림 4) 필터링을 위한 탐색 방법. 추정 거리의 하한값을 Key로 하는 우선순위 큐를 통해 탐색을 진행한다.

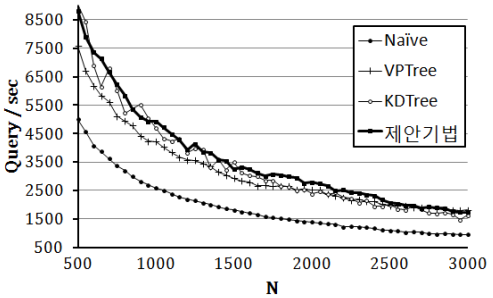
7. 실험 및 결과

실험을 위하여 변형 비속어 집합[2]을 데이터 크기 N에 따라 무작위로 선택하여 탐색 대상으로 사용하였고, 질의 단어로는 변형 비속어 단어를 포함하여, 랜덤 문자열, 한글 명사 사전에 수록된 일반 단어들을 이용하였다.

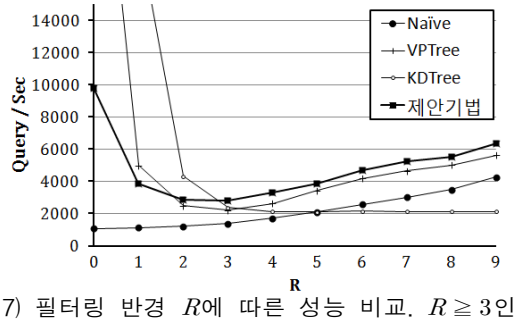
그림 5는 랜덤 문자열 쿼리에 대한 클러스터링 반경 ρ에 따른 초당 쿼리 처리량이다. 전체적으로 유사하지만 $\rho=0.1$일 때 가장 우수한 성능을 보인다. 그림 6은 데이터 크기 N에 따른 초당 쿼리 처리량(R=3)이다. 일부 구간에서 VP-Tree나 Kd-Tree가 우수한 구간이 일부 있지만 전체적으로 제안 기법이 우수한 것을 확인할 수 있다. 그림 7은 필터링 반경 R에 따른 초당 쿼리 처리량(N=2000)이다. 기본적인 탐색기법에 비해서는 항상 우수하며, 필터링 반경이 2 이상일 경우 VP-Tree보다, 반경이 3 이상인 경우에는 제안기법이 항상 우수함을 확인할 수 있다. 특히 Kd-Tree의 경우에는 필터링 반경이 증가할수록 성능이 급격하게 감소하는 것을 알 수 있다.



(그림 5) 군집화 매개변수 ρ에 따른 제안 기법의 성능. $\rho=0.1$일 때 가장 우수하다. (N=3000)



(그림 6) 데이터 크기 N에 따른 성능 비교. 전체적으로 제안기법이 우수한 것을 확인할 수 있다. (R=3)



(그림 7) 필터링 반경 R에 따른 성능 비교. $R \geq 3$인 경우 제안기법이 항상 우수한 것을 확인할 수 있다. (N=2000)

8. 결론 및 추후 연구

단어 필터링은 불건전한 정보를 차단하기 위한 필수적인 기능이다. 단어 필터링을 유연하게 수행하기 위해서는 우선 주어진 질의 단어와 일정 거리 미만으로 가까운 단어가 데이터베이스에 존재하는지를 탐색하는 근사 단어 필터링 탐색 문제를 해결해야 한다.

본 논문에서는 거리 공간을 만족하는 문자열간 거리 함수가 주어진 경우 효과적인 필터링 탐색을 수행할 수 있는 계층적 단어 군집화를 통한 상향식 자료구조 모델을 제안하였고 그 응용으로서 최소신장트리를 이용한 군집화 수행 기법을 이용하는 예를 소개하였다. 실험 결과 최소신장트리를 이용한 자료구조를 이용하여 필터링 탐색을 수행한 경우 데이터 크기 N=2000, 필터링 반경 R=3일 때, Kd-Tree에 비하여 16.9%, Vp-Tree에 비하여 26.6%의 성능 향상이 있음을 실험적으로 확인하였다.

다만, 본 논문에서 제안한 기법은 군집화 기법과 군집별 중심 단어 선정 방법, 거리 함수의 특성에 따라 성능의 차이가 존재할 것으로 판단되므로 군집화 기법의 종류 및 성능과 탐색 성능 간의 관계, 그리고 문자열 간의 거리 함수와 탐색 성능 간의 관계에 대한 보다 면밀한 실험적 연구가 추후 진행되어야 할 것이다.

감사의글

이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2011-0003157)

참고문헌

[1] 윤태진, 조환규, “반 전역 정렬을 이용한 온라인 게임 변형 욕설 필터링 시스템”, 한국콘텐츠학회논문지, 제9권 제12호, pp.113-120, 2009.

[2] 한국게임산업진흥원, 게임언어건전화지침서연구, 2008.

[3] 윤태진, 정우근, 조환규, “제한된 한글 입력환경을 위한 음소기반 근사 문자열 검색 시스템”, 정보과학회논문지:소프트웨어및응용, 제37권 제10호, pp.788-801, 2010.

[4] E. Chavez, G. Navarro, R. Baeza-Yates and J. L. Marroquin, "Searching in Metric Spaces," ACM Computing Surveys, Vol.33, No.3, pp.273-321, 2001.

[5] 윤태진, 조환규, “계층적 메트릭 공간 구조의 한글 근사 단어 검색 시스템”, 제33회 정보처리학회 춘계학술발표대회논문집, 제17권 제1호, pp.397-400, 2010

[6] S. C. Sahinalp and M. Tasan, "Distance Based Indexing for String Proximity Search", in Proc. of Int'l Conf. on Data Engineering,

[7] B. Bustos, G. Navarro and E. Chavez, "Pivot Selection Techniques for Proximity Searching in Metric Spaces," Pattern Recognition Letters, Vol.24, No.14, pp.2357-2366, 2003.