Approximate Searches

Similarity searches in Postgresql using metric spaces

Enrico Pirozzi
www.psql.it
scotty@psql.it
info@enricopirozzi.info

PGCON 2008 - OTTAWA

We are going to talk about

- Searches with exact match
- Metric Spaces
- Approximate Searches
- Edit Distance
- Example
- Pivoting - Indexing
- Future issues

Requirements to Approx searches

- Postgresql
- C language
- SQL language
- Pl-pgsql language

Exact Match

Select * from customers where name = 'Enrico'

Result :

name	address	town
Enrico	*2, red street*	*Ottawa*

There is match!!

Exact Search : when can I use it?

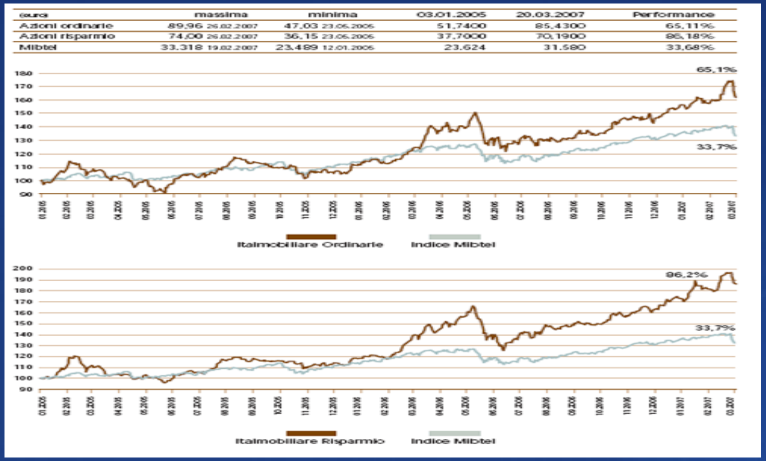
It make sense to use an exact search when we presume that our result is inside our database -> .....where name='Enrico'

EXACT SEARCH

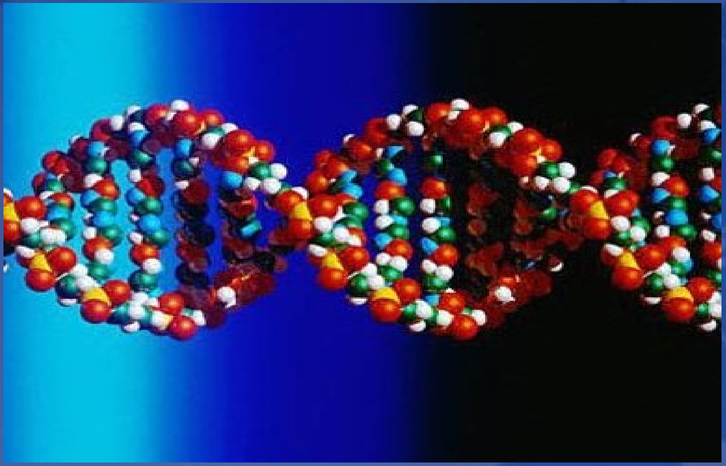
Other kind of search ?

Images Search

Sounds Search

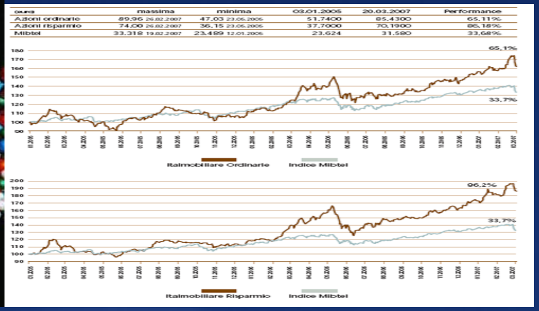


Economic Search



DNA Search

Other kind of search ?



Similarity Searches

Similarity Searches

Some Examples:

- Text retrievial
- Image searches
- Sound searches
- Other .....

Similarity Searches

	id [PK] serial	nome character var
1	1	pippo
2	2	pluto
*		

In the exact searches we think our dataset as a set of rows and we make searches inside it

When we make a similatity search we have to think our dataset as a set of objects. The query is an object too that can belong or not belog to the dataset

Similarity Searches

The *“similiarity search”* is the search of objects that belong to the dataset and that are closer to the query object

Edit distance

Edit distance or Levenshtein distance:

The Edit distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character.

Edit distance

For example the distance between "kitten" and "sitting" is 3

- kitten → sitten (substitution of 's' for 'k')
- sitten → sittin (substitution of 'i' for 'e')
- sittin → sitting (insert 'g' at the end)

Edit distance

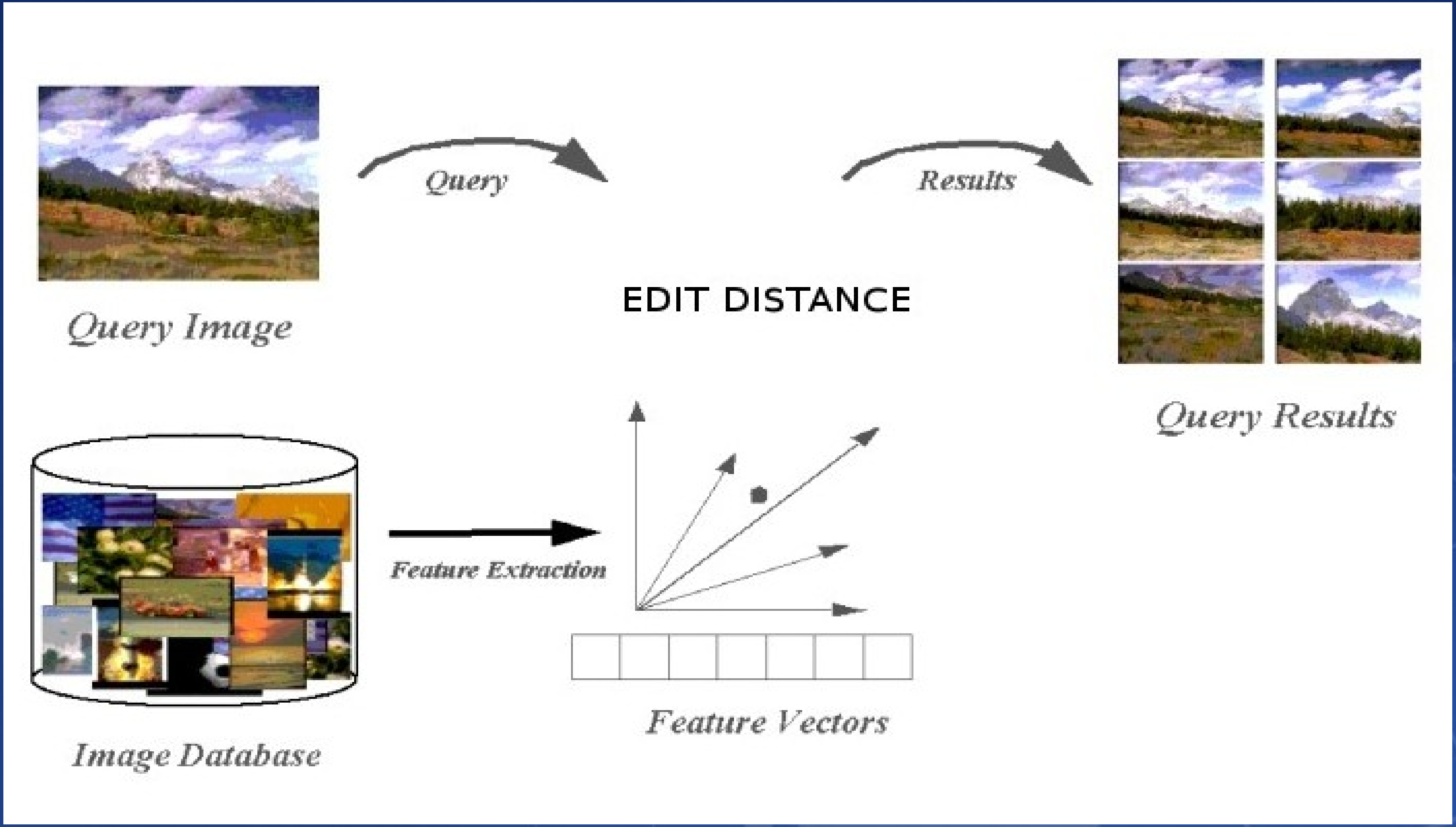
Now we can compare:

- Strings

- Every kind of object starting from its features -> Every object has features

- We find an object that has the smallest distance from our query.

Edit distance



Some Problems

Objects can have many dimensions : tipical 50-60 dimensions 1 dimension for each feature (for example 3 for RGB images, and so on...)

We can spend much time to calculate the edit distance.

Metric Spaces

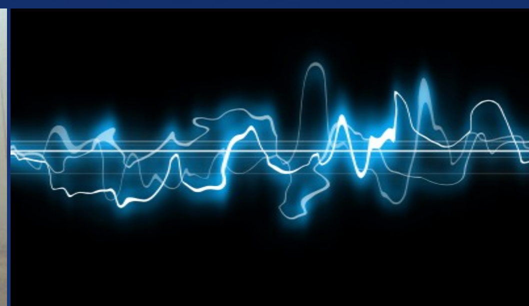
A metric space is a set S with a global distance function (the metric d) so that for every two points x,y in S, returns the distance between them as a nonnegative real number d(x,y).

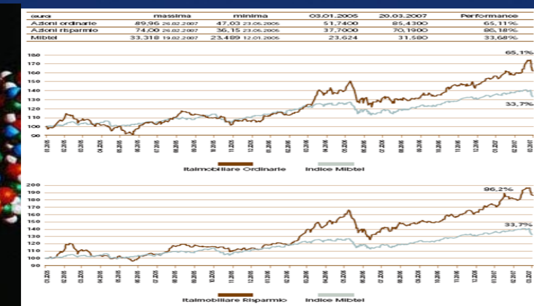
Metric Spaces

The distance function d(x,y) must be :

- non-negative: d(x,y)>=0
- Strictly Positive : d(x,y)=0 iff x=y
- Symmetric: d(x,y)=d(y,x)
- Have to satisfy the triangle inequality : d(x,z)<=d(x,y)+d(y,z)

Metric Spaces



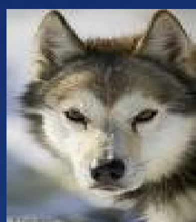
Database objects are seen as points in a metric space

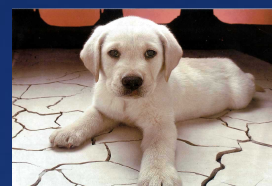
Query point can belong to the dataset

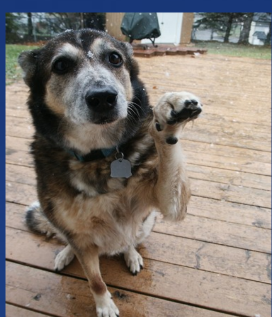
Multimedia Dataset

Query

Query point can not belong to the dataset



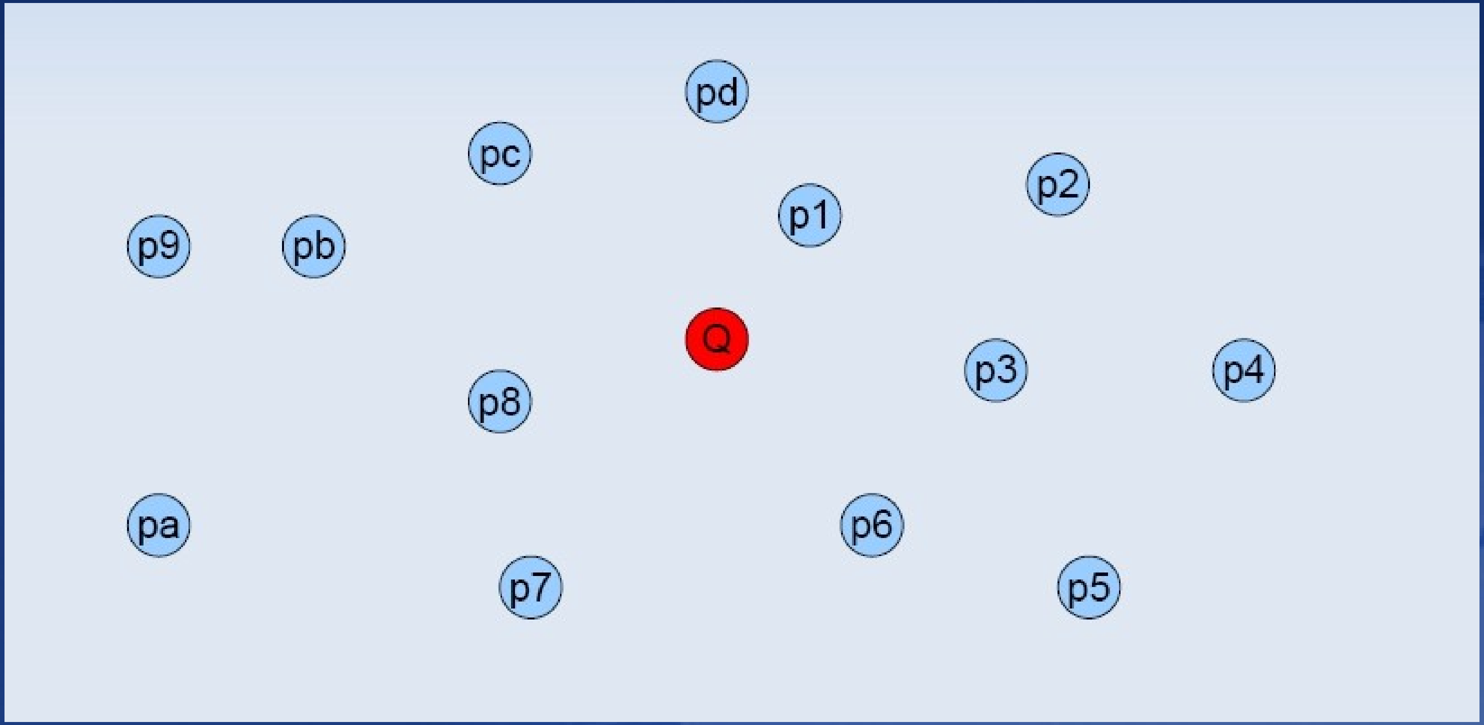




Query

Multimedia Dactaset

Dataset vs points



Objects and queries are seen as points in a multidimensional metric space

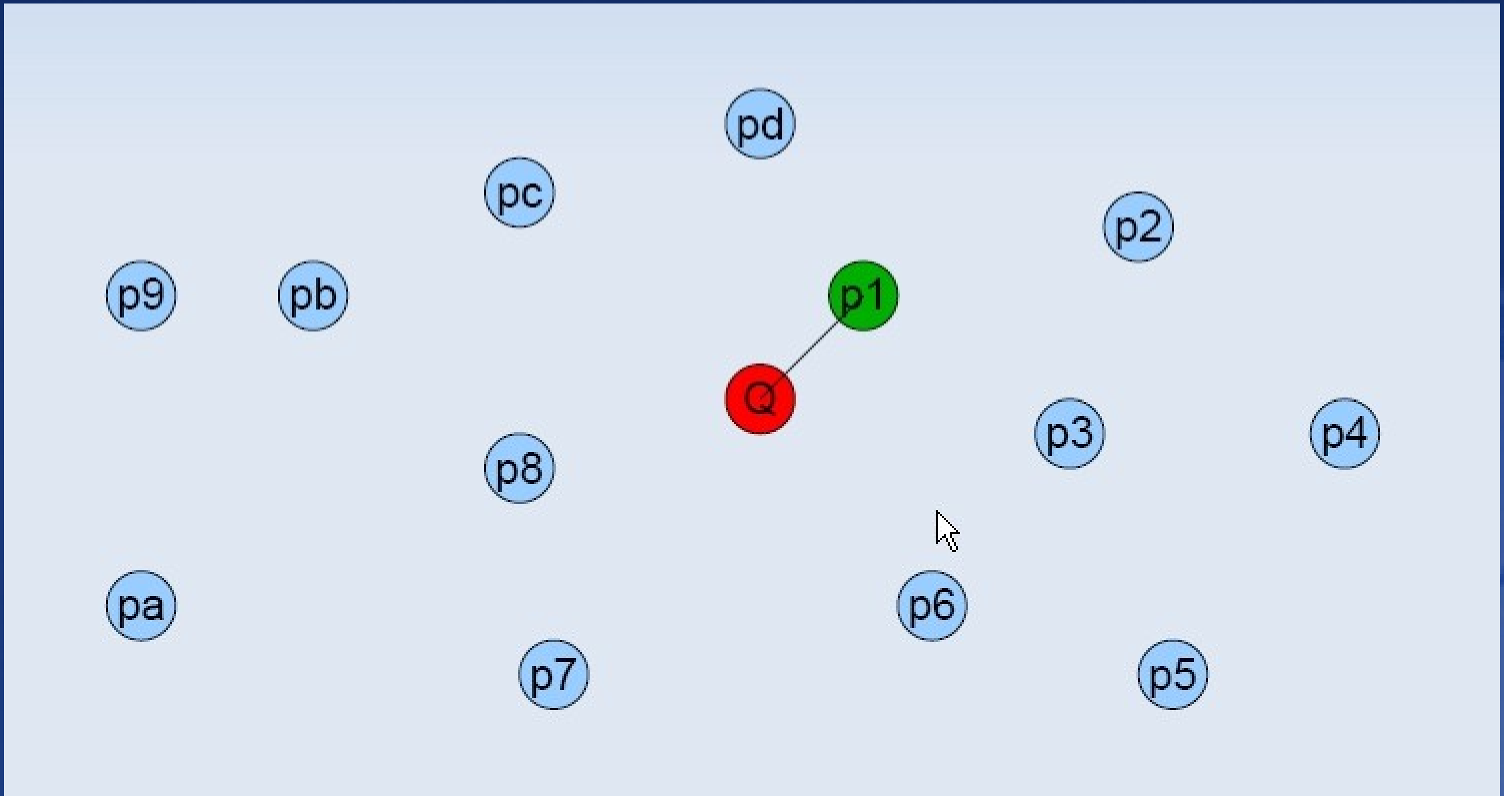
Metric Spaces: Searches

- Nearest neighbor : search of the object more close to the query point

- K - Nearest neighbor : search of the K objects more close to the query point

- Range query : search of objects that are inside the circle with a given radius r and center in query point q

Nearest Neighbor



The point p1 is the nearest neighbor for the query point q

K – Nearest Neighboor: an example

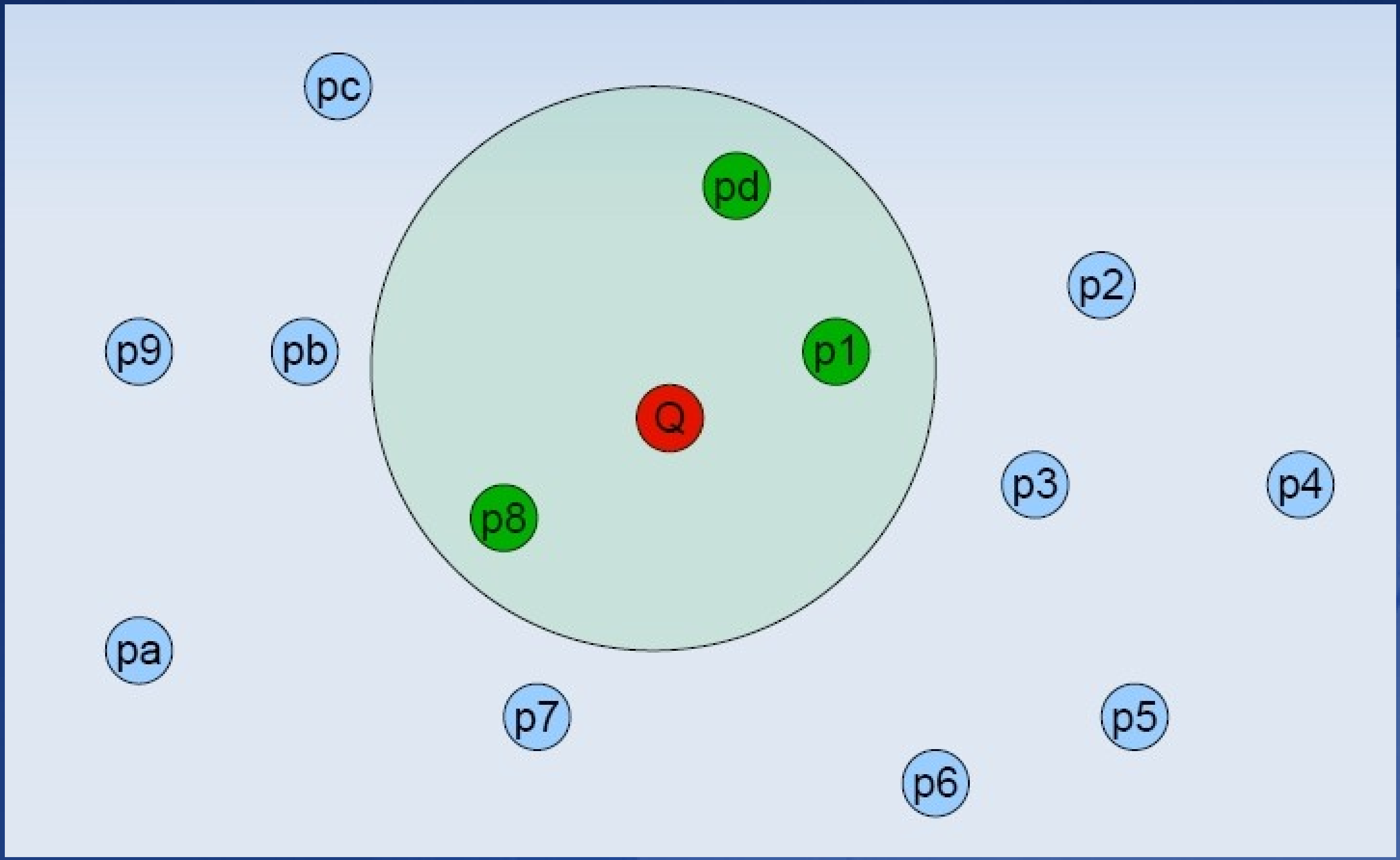
```
SELECT parola, editdistance  (parola,
'contorno') from parole order by 2 limit 5;
```

quattro
terre
comparsa
donna
scomparso
nonno
volto
quattro

contorno

```
parola    |  editdistance
----------------------------
nonno     |  4
comparsa  |  5
donna     |  5
volto     |  5
quattro   |  5
```

Range Query



Points p1,p8,pd are inside the circle

Range Query : an example

```
SELECT parola, editdistance  (parola,
'contorno') from parole where
editdistance(parola,'contorno') < = 5;
```

quattro
terre
comparsa
donna
scomparso
nonno
volto
quattro

contorno

```
parola    |  editdistance
---------------------------------------
nonno     |  4
comparsa  |  5
quattro   |  5
volto     |  5
donna     |  5
```

Similarity searches

- Instead of words we can use any kind of strings
- We can compare n-ple of values(a1,a2,..., an) that represents features of objects.

It is possible to make similiarity searches between any kind of objects

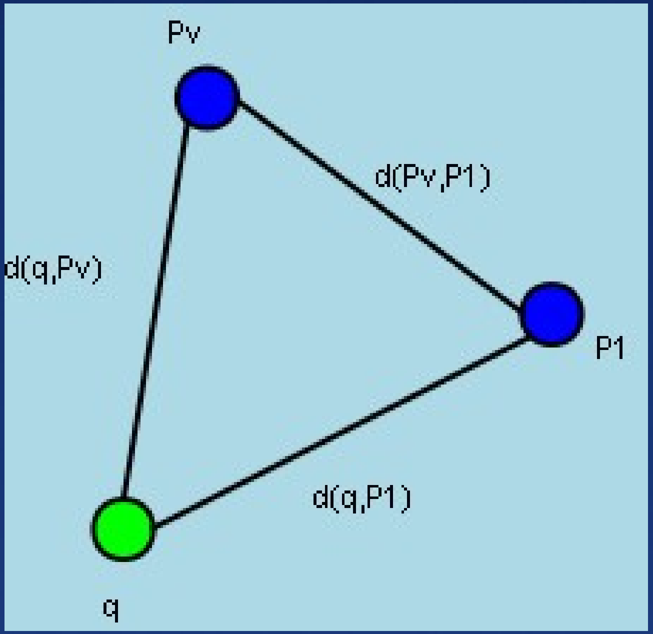
The dark side

- High dimensional spaces tipical 50,60 dimensions for each object

Problems:

- A lot of memory
- A lot of time to calculate edit distance

The triangle inequality



$$d(q,p1) <= d(q,Pv)+d(Pv,P1)$$

The triangle inequality

Let (X,d) be a metric space, where X is the universe of valid objects and d is the metric of the space, and let U a subset of objects of X |U|=n U is our database.

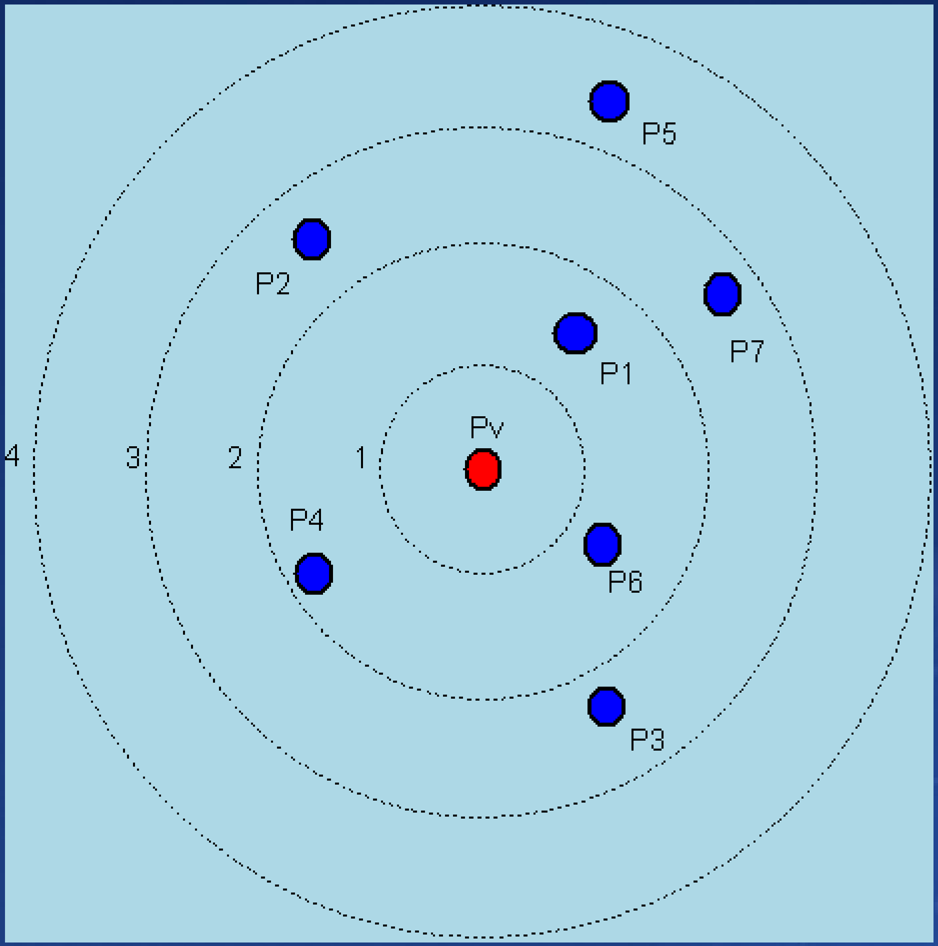
(q,r) = {u that belongs to U so that d(u,q)<=r}   Range Query

Given a quey (q,r) and a set of k pivots {p1,...,pk} by the triangle inequality it follows that d(pi,x)<= d(pi,q)+q(q,x), and also that d(pi,q)<= d(pi,x)+d(x,q) for any x that belongs to X. From both inequalities, it follows that a lower bound on d(q,x) is d(q,x) >= |d(pi,x)-d(pi,q)|. The objects u of interest are those that satisfy d(q,u)<=r, so all the objects that satisfy the exclusion condition can be excluded, without actually evaluating d(q,u)

| d(pi,u)-d(pi,q) | > r for some pivots p_i

Do you need some coffee ?

Building an Index - Pivoting

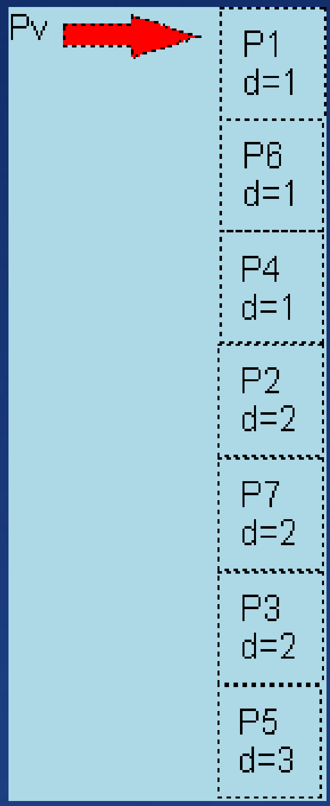


Some examples:

d(Pv,P4)=1
d(Pv,P3)=2
..... and so on

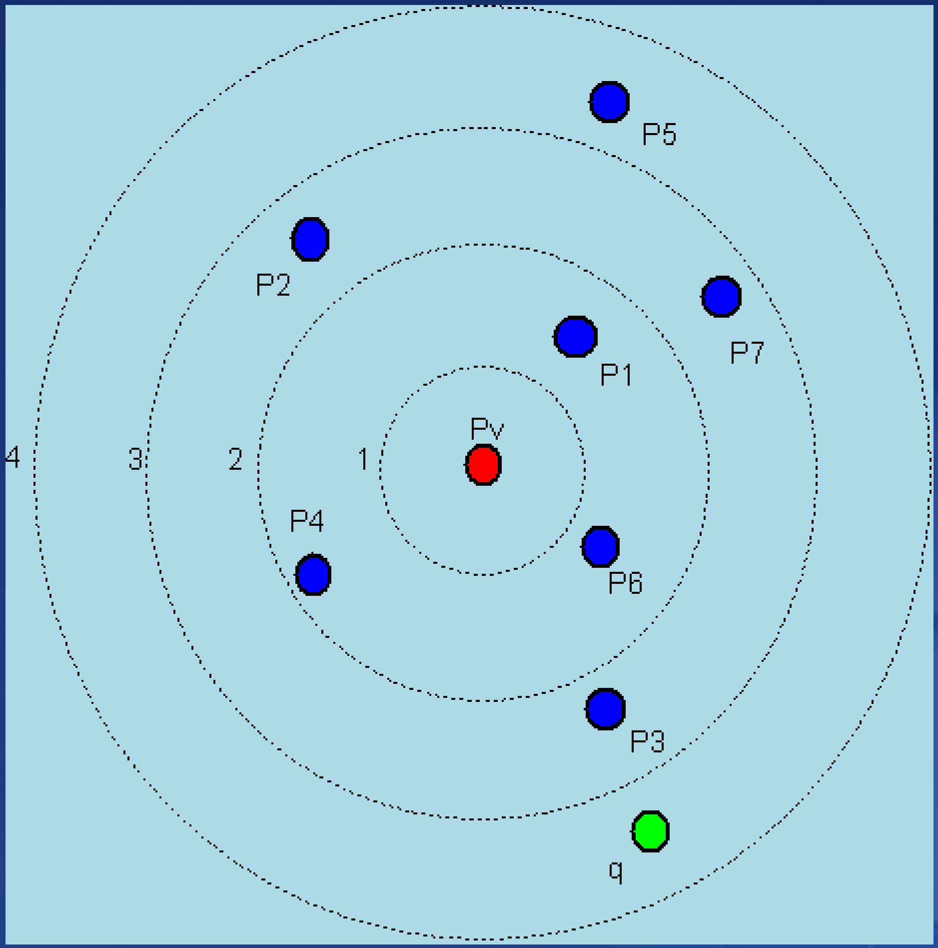
We can choose a point as pivot point and when we insert a new item we can pre-calculate the distance beetween our pivot and the new point

Building an Index - Pivoting



We Can store our informations in an index ordered by distance between the pivot and the other point of the metric space

Building an Index – Range Query

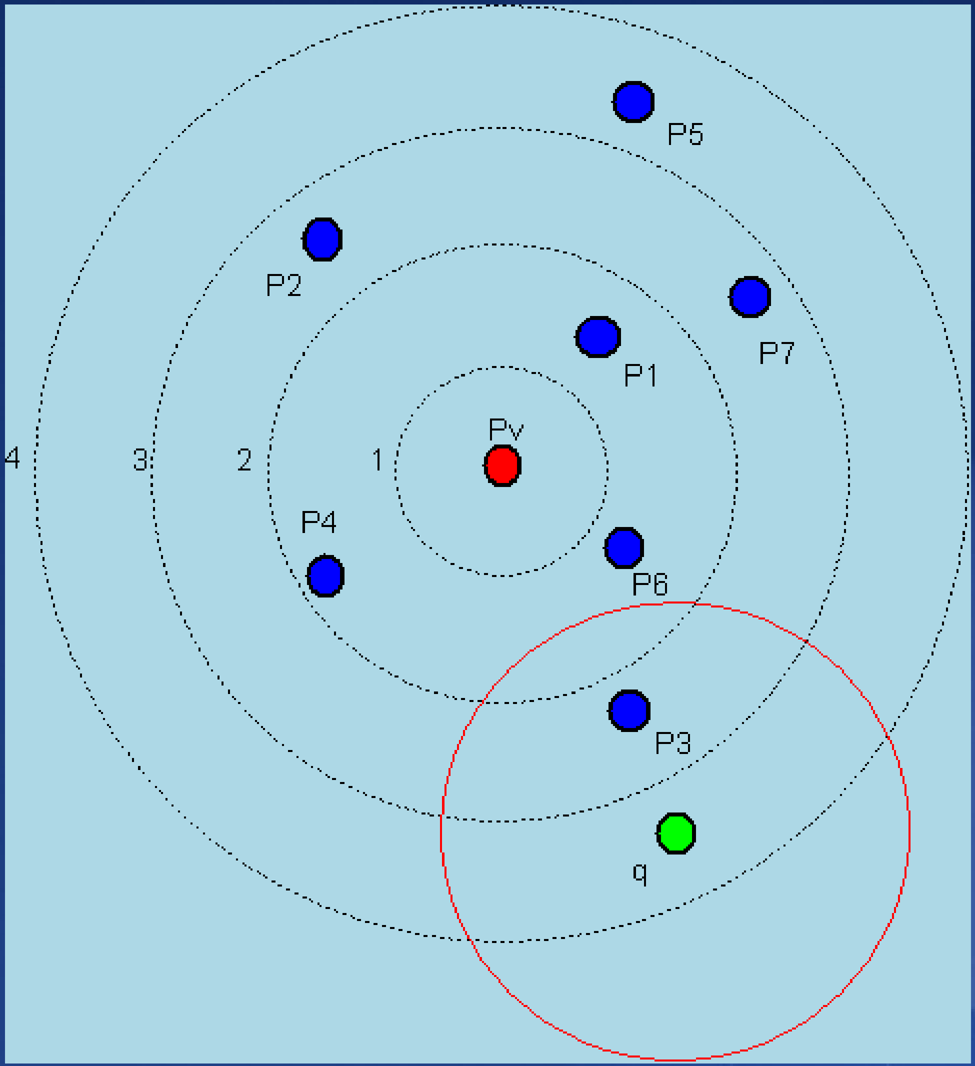


Q is our query point

We want to find all the points such that

d(q,pi) < = 1

Building an Index – Range Query

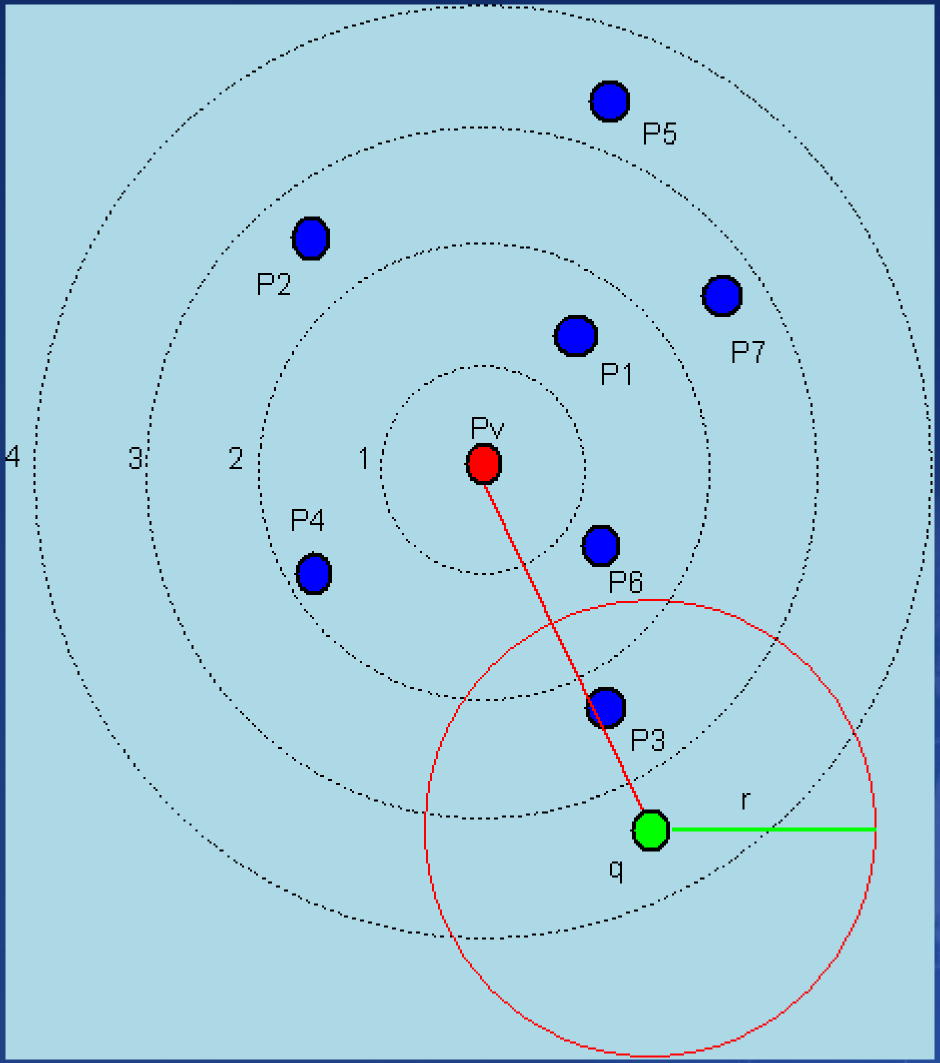


Q is our query point

We want to find all the points such that

$d(q,pi) <= 1$

Building an Index – Range Query

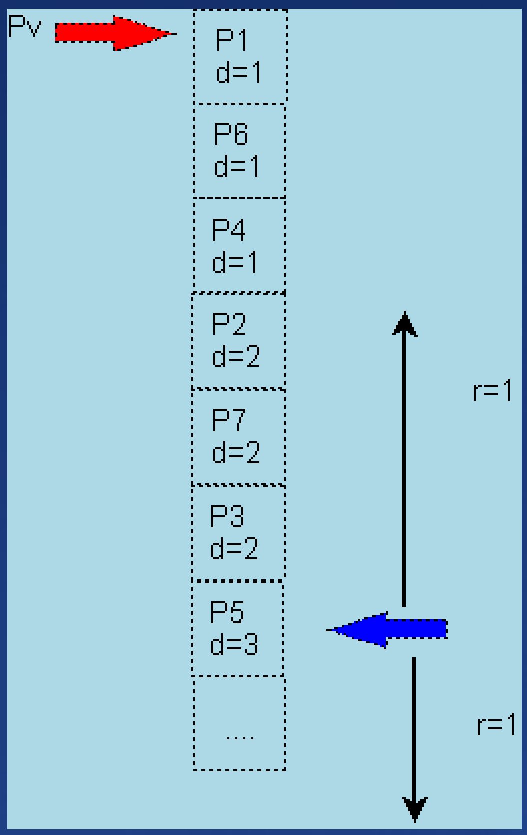


d(q,pv) = 3

r=1

d(Pv,Pi)-d(q,Pi)<=1

Building an Index – Range Query



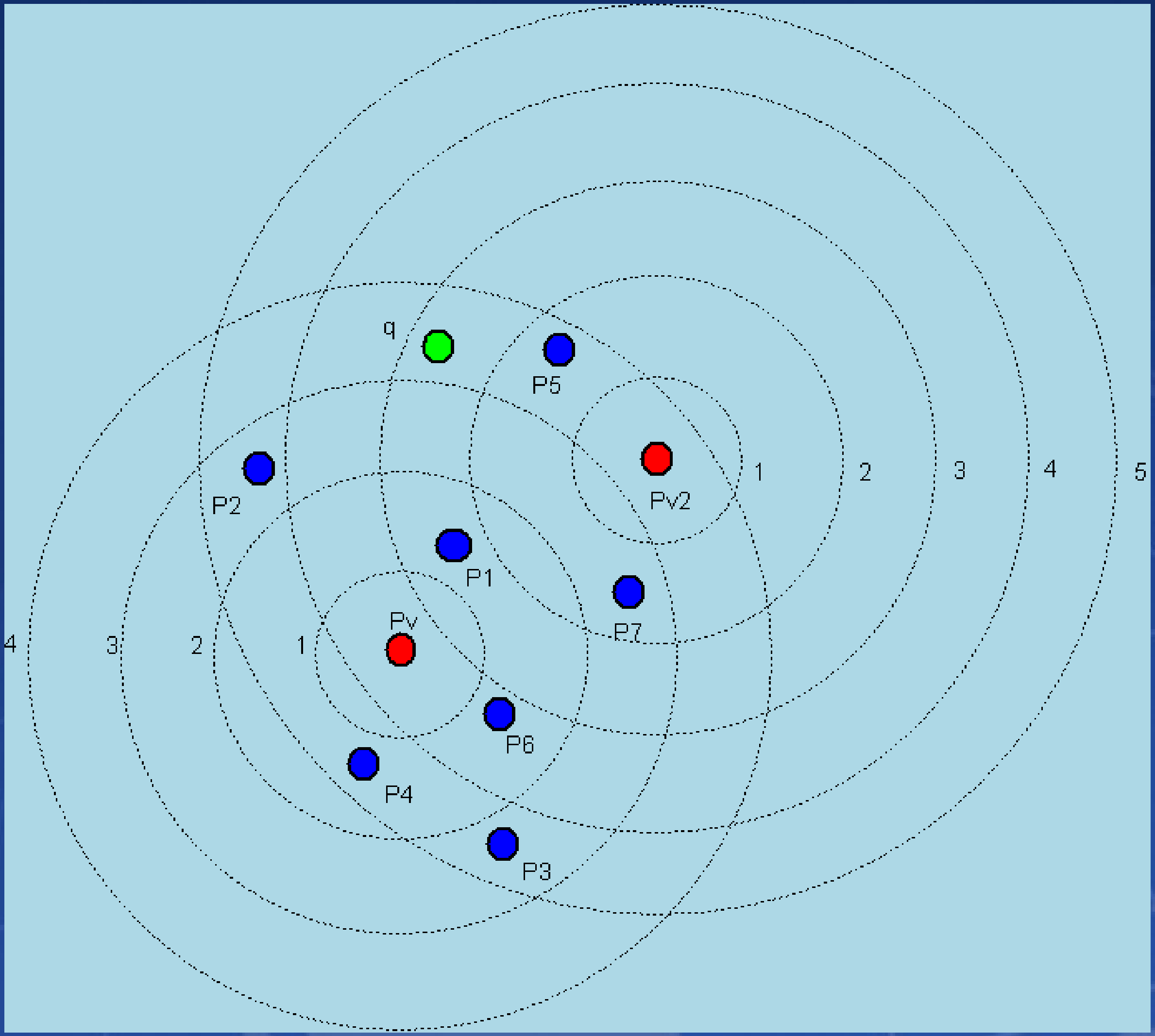
We center our search in d(Pv,q) = 3 and choose our cadidates point inside the interval between [d(Pv,q)-r , d(Pv+r)]

Our candidates are P2,P7,P3,P5

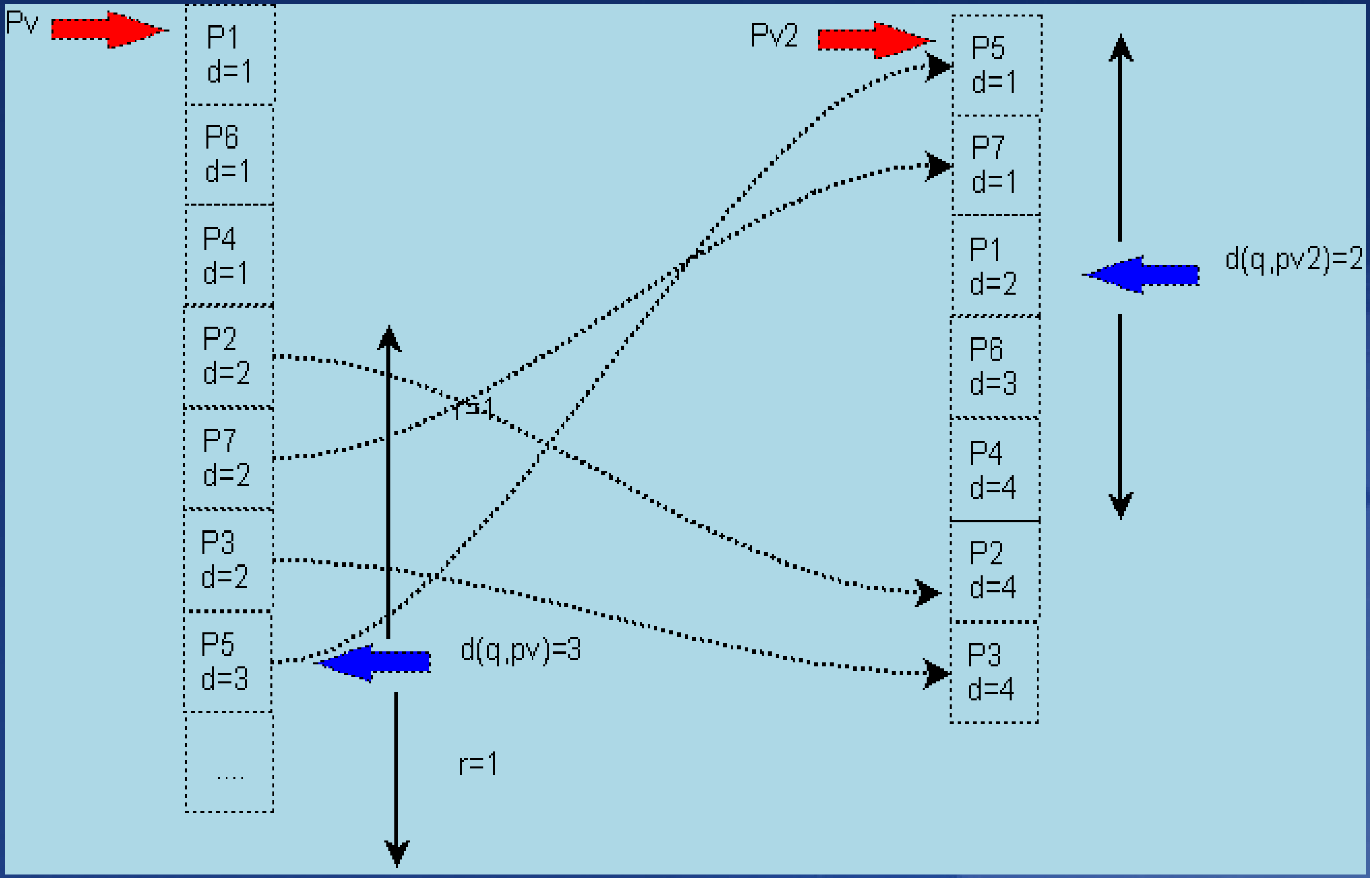
After calculating the edit distance among those points and q we will see that P3 is in the result set

Building an Index - 2 Pivots

If we have 2 or more pivots we consider as candidate points all the point that are in the intersections of the distance calculated among pivots



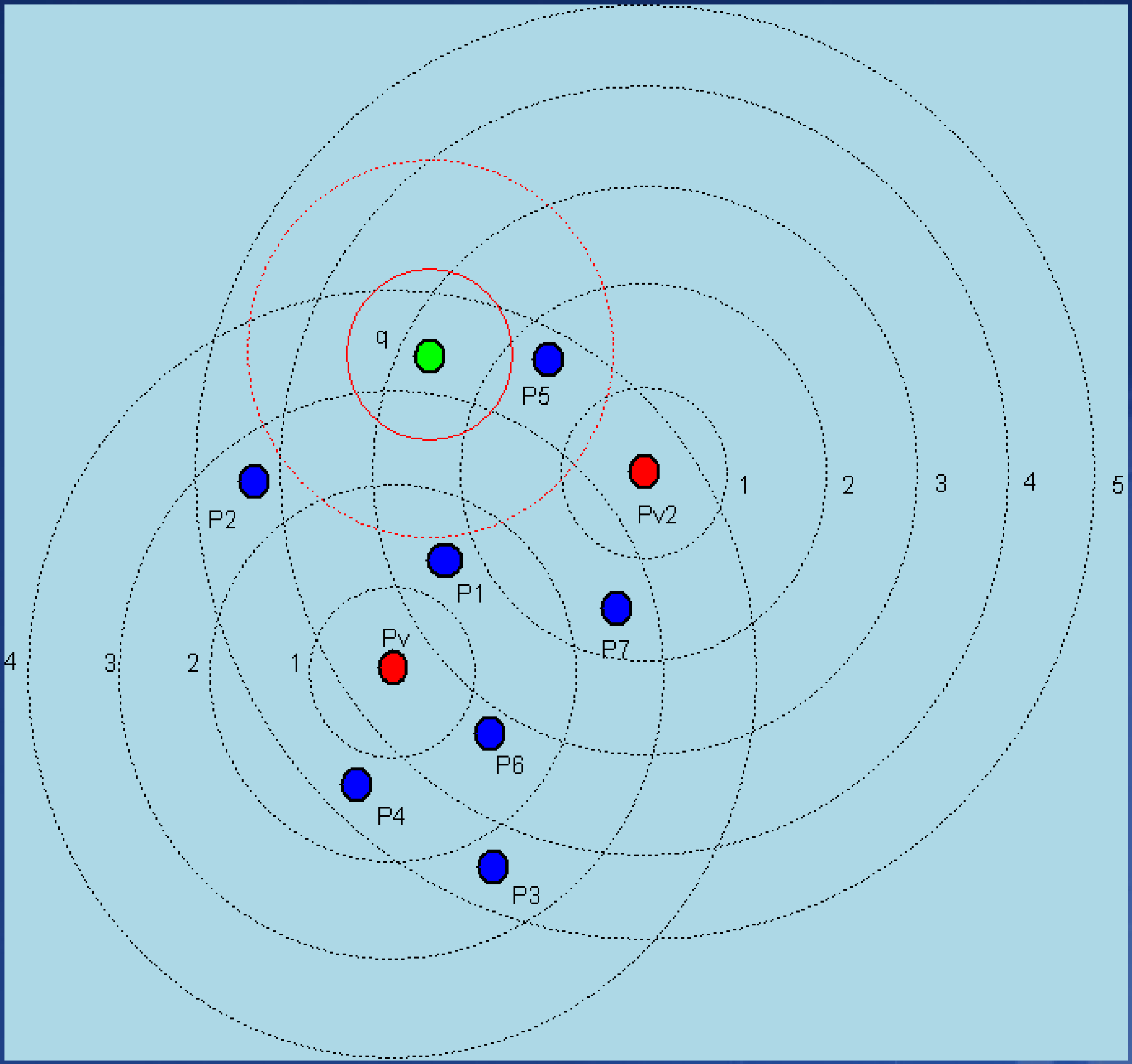
Building an Index - 2 Pivots



P5 and P7 belong to the intersection

P5 and P7 are the candidates point

Building an Index - 2 Pivots

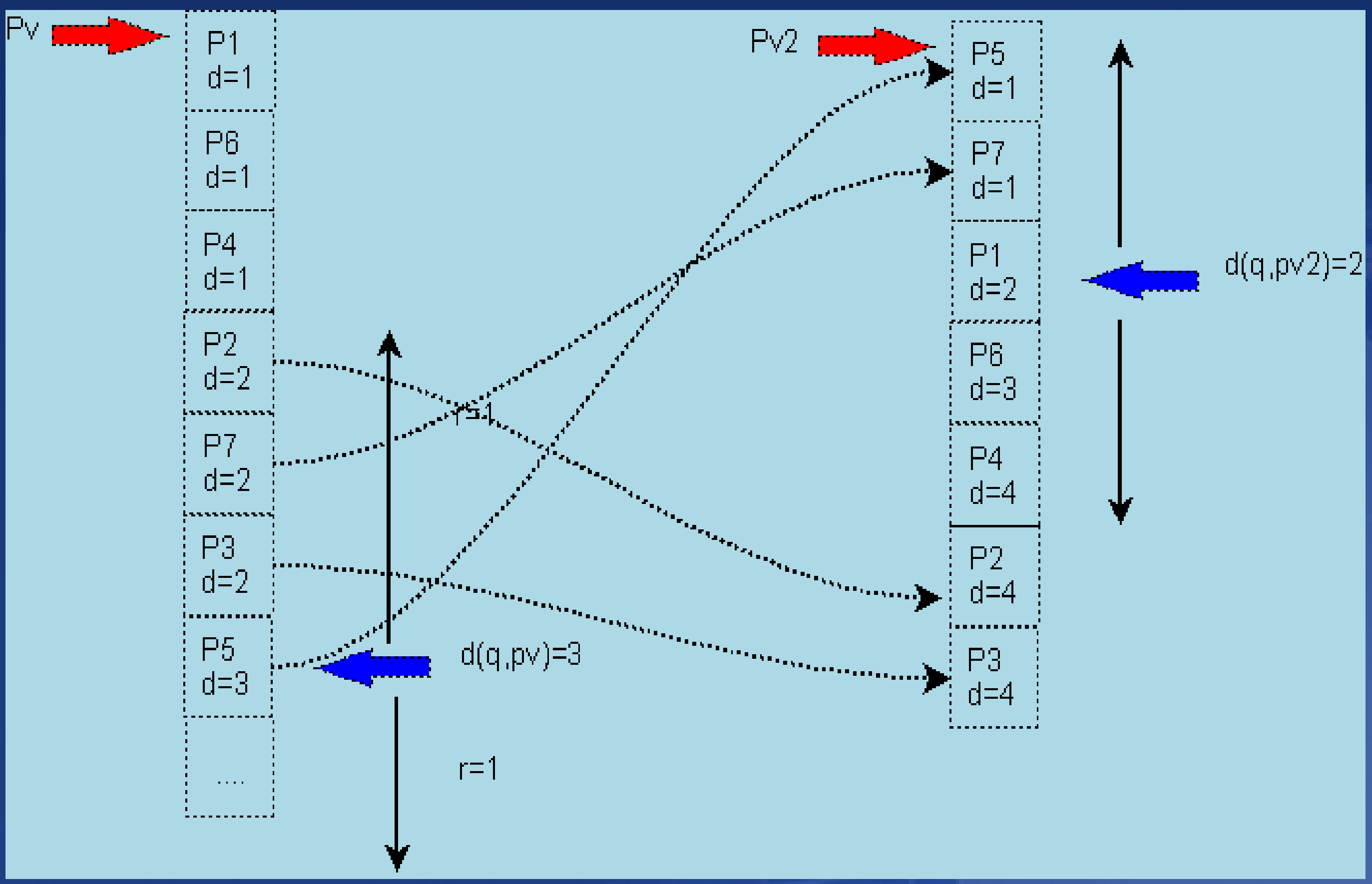


An example:

Range query
query point q
radius r = 2

P5 belong to the circle
with center q and radius r

Building an Index - 2 Pivots



P5 is a cadidate point and

P5 is inside the circle with center q and radius r

Other features

- We can implement an algoritm also for K-NN queries using our index structure
- We can use an approximated edit-distance function (using AC or PAC alghoritms) to minimize computational time.

Status of work

APPROXIMATED SEARCHES

EDIT DISTANCE

On Pg-foundry contrib Pg-edist

INDEX FOR METRIC SPACES TO IMPROVE RANGE QUERIES AND K-NN QUERIES

It already works, but it is written in C Language and it is not yet present on PostgreSQL

The Future, but ....

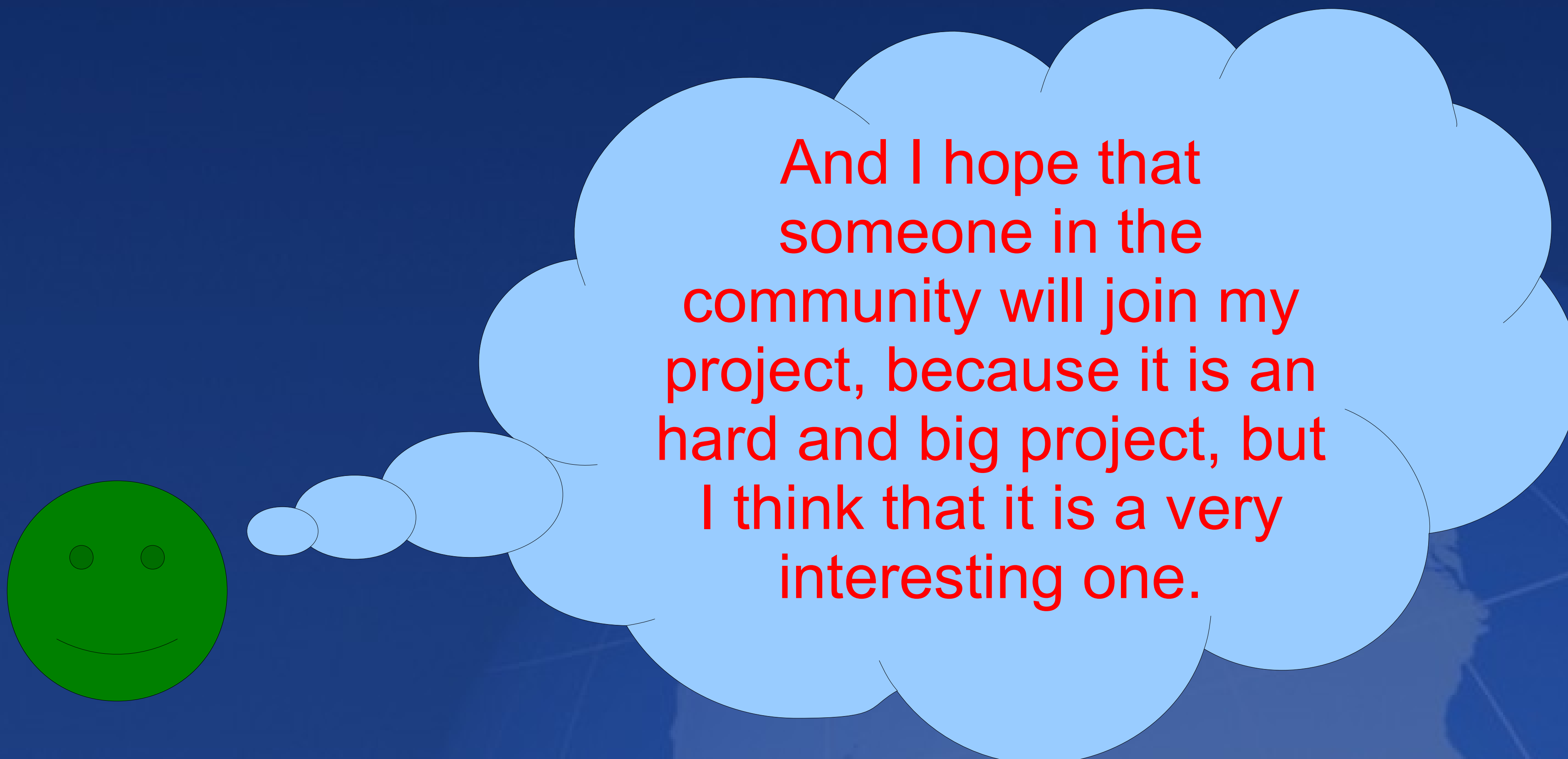
APPROXIMATED SEARCHES

EDIT DISTANCE

INDEX FOR METRIC SPACES
TO IMPROVE RANGE QUERIES AND K-NN QUERIES

I want to extend Pg-edist with the C code written about the index structure

................................I'm alone


And I hope that someone in the community will join my project, because it is an hard and big project, but I think that it is a very interesting one.

The End: we talked about

- Metric Spaces
- Approximate Searches
- Edit Distance
- Example
- Pivoting - Indexing
- Future issues

Bibliography

- My thesis : http://www.enricopirozzi.info
- Searching in metric spaces
  http://citeseer.ist.psu.edu/avez99searching.html
- Spaghettis: An Array Based Algorithm for Similarity Queries in Metric Spaces
  http://citeseer.ist.psu.edu/414510.html
- Overcoming the Curse of Dimensionality
  http://citeseer.ist.psu.edu/407814.html
- Fixed Queries Array: A Fast and Economical Data Structure for Proximity Searching
  http://citeseer.ist.psu.edu/ch01fixed.html

Contact Information

- Web Site: www.psql.it
- Email: scotty@psql.it
- Personal pages: www.enricopirozzi.info
- Email: info@enricopirozzi.info
- Skype contact: sscotty71
- Gtalk contact: sscotty71@gmail.com

License