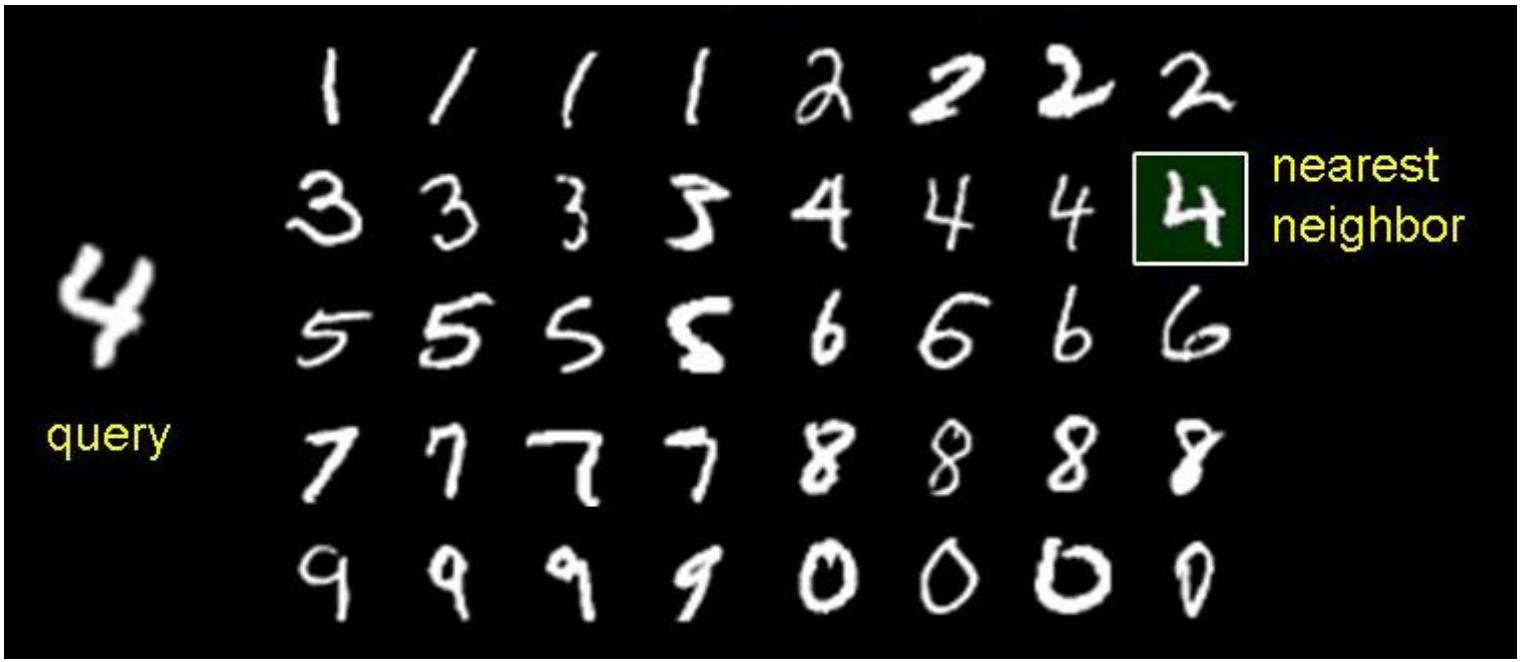
Offline nearest neighbors

I.Z. Emiris, I. Psarros and O. Rouillé

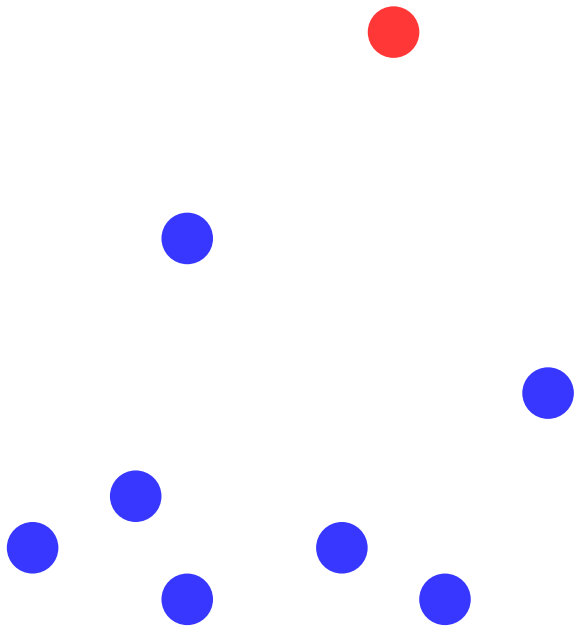
Department of Informatics & Telecommunications
National and Kapodistrian University of Athens, Greece

March 30, 2017

Classification

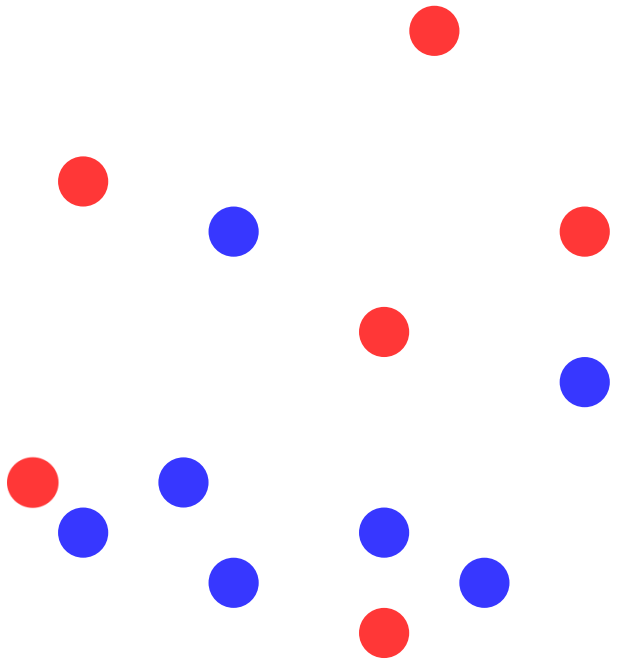


Offline nearest neighbors



The Nearest Neighbor problem is to find the nearest point (blue) to a given query (red).

Offline nearest neighbors



The Offline Nearest Neighbors problem is to find, for each query, the nearest point.

Offline nearest neighbors: formal definition

The Offline Nearest Neighbors problem

Given a distance d and two sets of points, the queries R and the data B, the Offline Nearest Neighbors problem is to find the closest point to each query. In other words, for all $r \in R$, find $b \in B$ such that $\forall b' \in B,\ d(r, b) \leq d(r, b')$.

Offline nearest neighbors: formal definition

The Offline Nearest Neighbors problem

Given a distance d and two sets of points, the queries R and the data B, the Offline Nearest Neighbors problem is to find the closest point to each query. In other words, for all $r \in R$, find $b \in B$ such that $\forall b' \in B,\ d(r, b) \leq d(r, b')$.

Here we actually deal with the approximate offline nearest neighbors: let $\epsilon > 0$ fixed, for all $r \in R$, find $b \in B$ such that $\forall b' \in B,\ d(r, b) \leq (1 + \epsilon)d(r, b')$.

Offline nearest neighbors: strategy

Offline nearest neighbors takes advantage from the fact that all the queries are known in advance.
We aimed for solutions that:

- use the fact that we know the queries in advance;
- were not exponential in the dimension.

We worked on the problem by applying different algorithms/methods in C++ on an existing benchmark.

Plan

Locality Sensitive Hashing

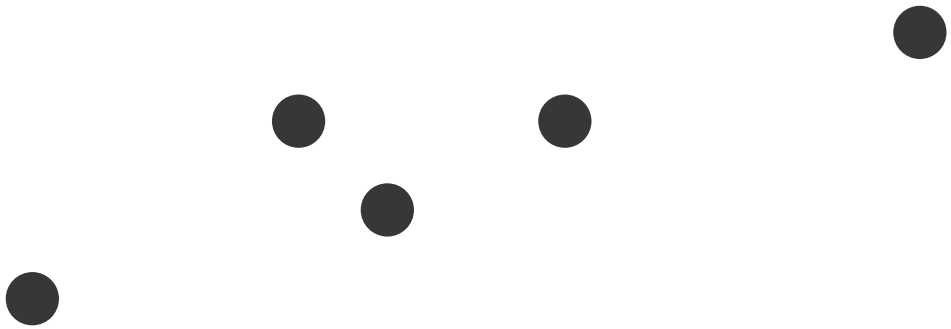
Definition

For a metric space (M,d) a threshold $R > 0$, an approximation factor $c > 1$ and two probabilities $P_2 < P_1$, a LSH-function is a function h such that $\forall (p, q) \in M^2$,
$(d(p, q) \leq R \Rightarrow P[h(p) = h(q)] \geq P_1) \wedge$
$(d(p, q) \geq cR \Rightarrow P[h(p) = h(q)] \leq P_2)$.

Locality Sensitive Hashing

Definition

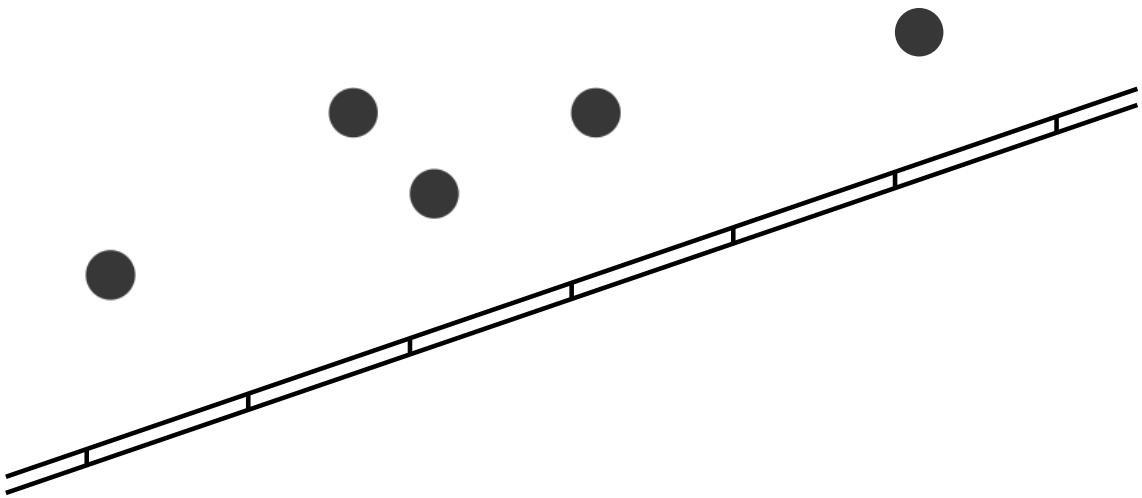
For a metric space (M,d) a threshold $R > 0$, an approximation factor $c > 1$ and two probabilities $P_2 < P_1$, a LSH-function is a function h such that $\forall (p, q) \in M^2$,
$(d(p, q) \leq R \Rightarrow P[h(p) = h(q)] \geq P_1) \wedge$
$(d(p, q) \geq cR \Rightarrow P[h(p) = h(q)] \leq P_2)$.



Locality Sensitive Hashing

Definition

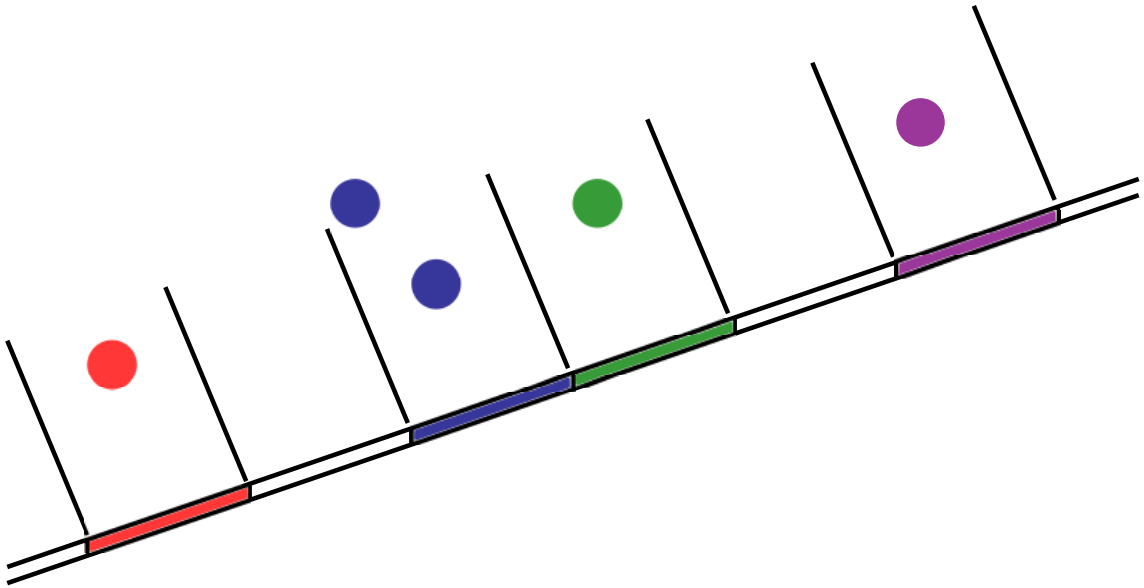
For a metric space (M,d) a threshold $R > 0$, an approximation factor $c > 1$ and two probabilities $P_2 < P_1$, a LSH-function is a function h such that $\forall (p, q) \in M^2$,
$(d(p, q) \leq R \Rightarrow P[h(p) = h(q)] \geq P_1) \wedge$
$(d(p, q) \geq cR \Rightarrow P[h(p) = h(q)] \leq P_2)$.



Locality Sensitive Hashing

Definition

For a metric space (M,d) a threshold $R > 0$, an approximation factor $c > 1$ and two probabilities $P_2 < P_1$, a LSH-function is a function h such that $\forall (p, q) \in M^2$,
$(d(p, q) \leq R \Rightarrow P[h(p) = h(q)] \geq P_1) \wedge$
$(d(p, q) \geq cR \Rightarrow P[h(p) = h(q)] \leq P_2)$.



How does Dolphinn work?

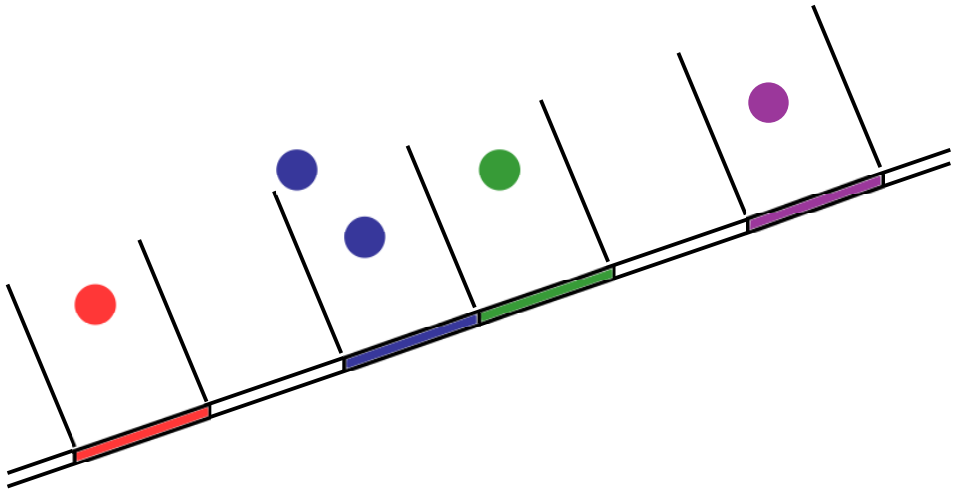
Dolphinn is an LSH-based algorithm used for the near neighbors search.

How does Dolphinn work?

Dolphinn is an LSH-based algorithm used for the near neighbors search.
It uses a family of LSH functions and reduces to two the size of their codomain.

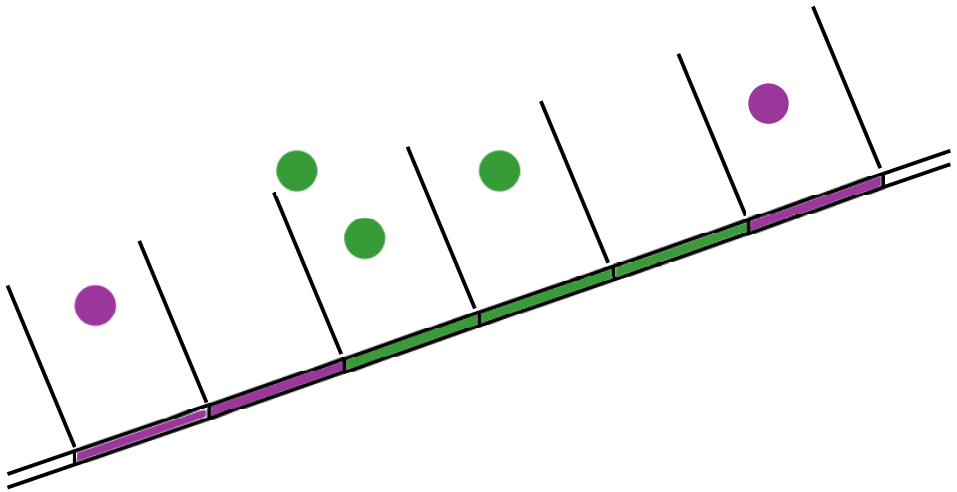
How does Dolphinn work?

Dolphinn is an LSH-based algorithm used for the near neighbors search.
It uses a family of LSH functions and reduces to two the size of their codomain.



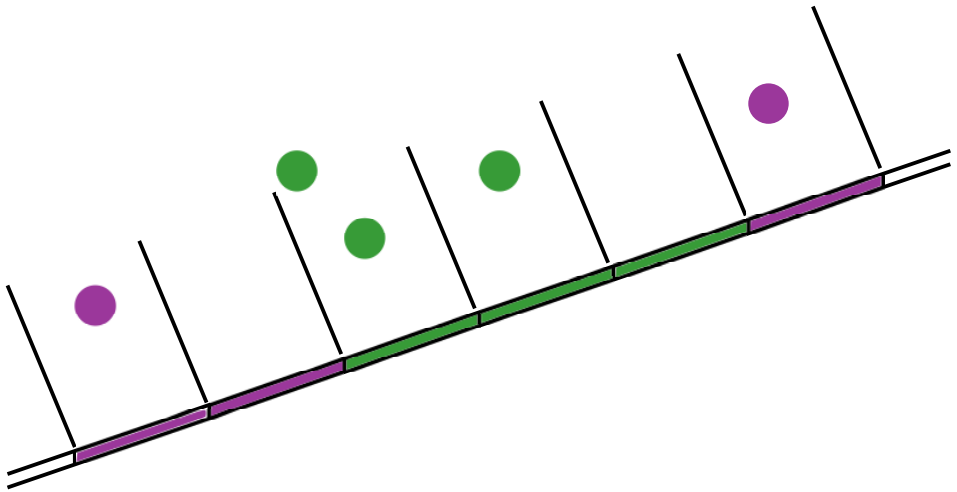
How does Dolphinn work?

Dolphinn is an LSH-based algorithm used for the near neighbors search.
It uses a family of LSH functions and reduces to two the size of their codomain.



How does Dolphinn work?

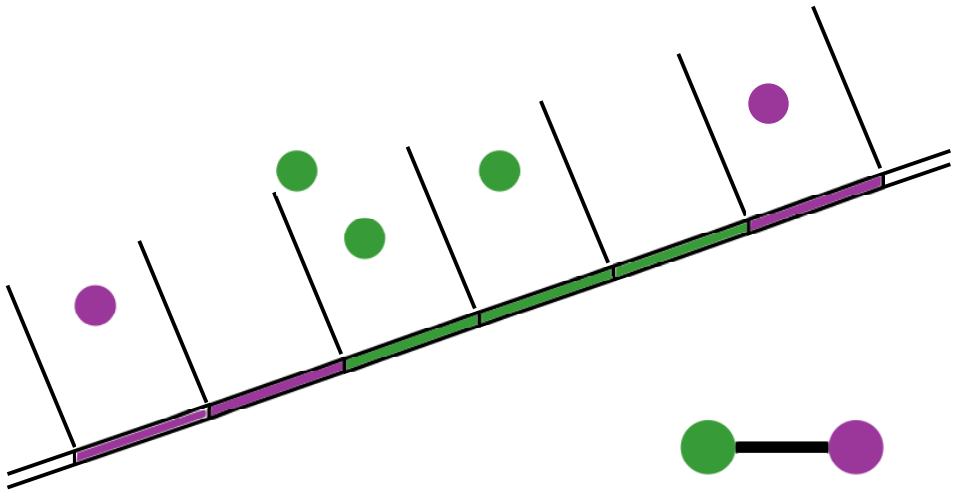
Dolphinn is an LSH-based algorithm used for the near neighbors search.
It uses a family of LSH functions and reduces to two the size of their codomain.



The image through such an LSH function holds on a bit (0 or 1), the image through the family provides a vector of 0 and 1.

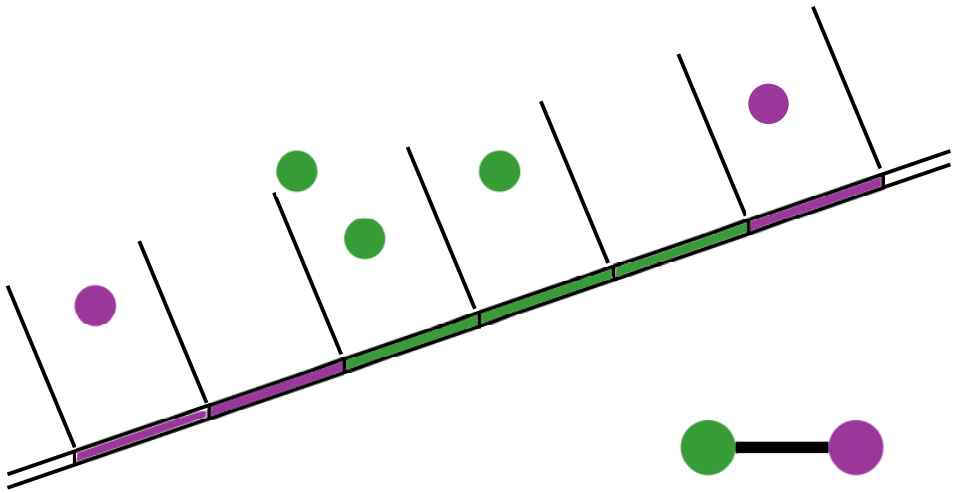
How does Dolphinn work?

Dolphinn is an LSH-based algorithm used for the near neighbors search.
It uses a family of LSH functions and reduces to two the size of their codomain.



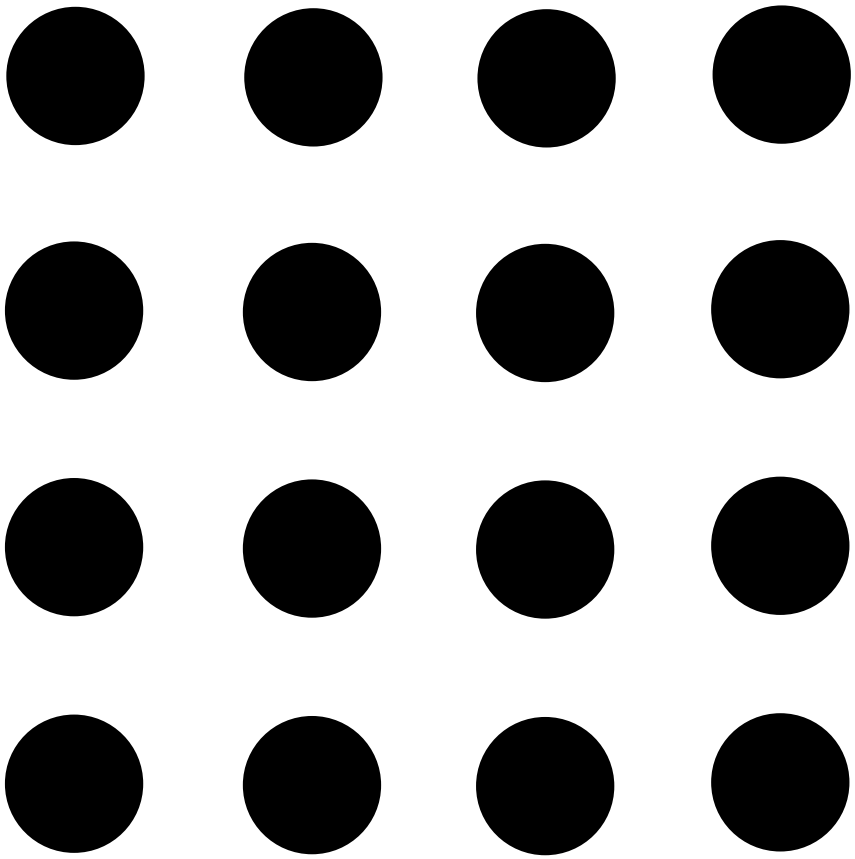
How does Dolphinn work?

Dolphinn is an LSH-based algorithm used for the near neighbors search.
It uses a family of LSH functions and reduces to two the size of their codomain.

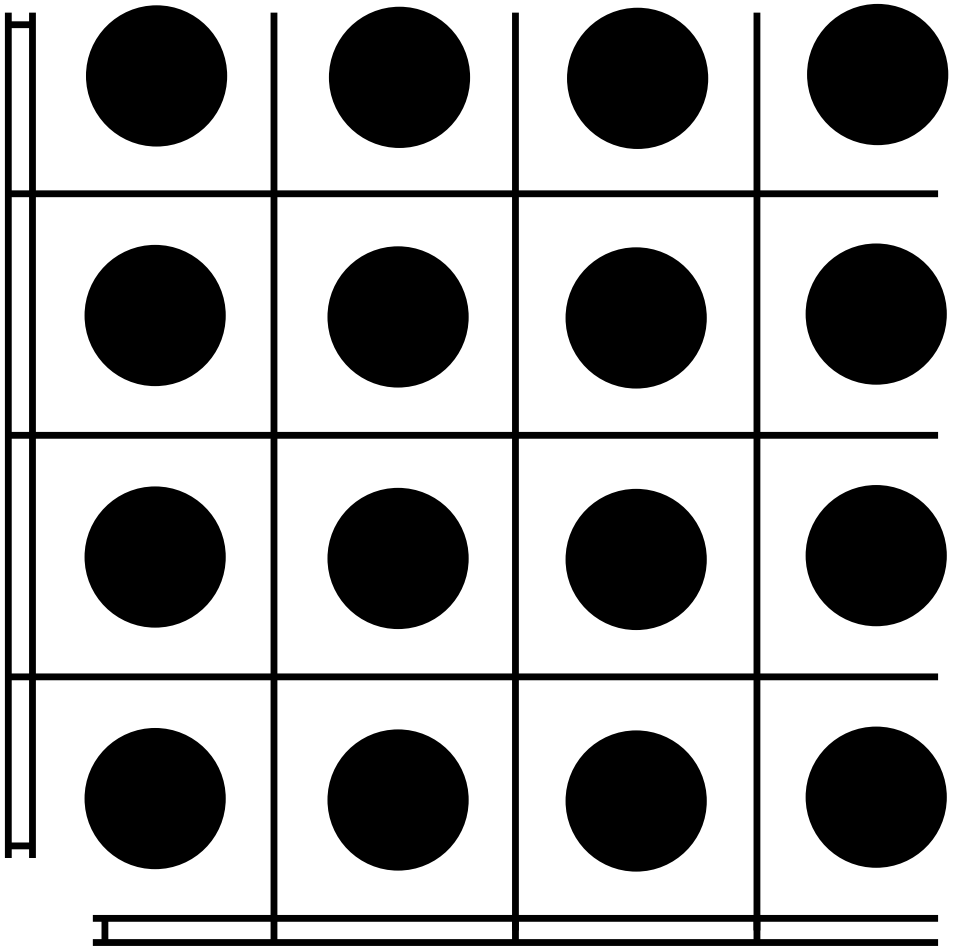


With the LSH family, Dolphinn produces a hypercube of dimension the number of LSH functions.

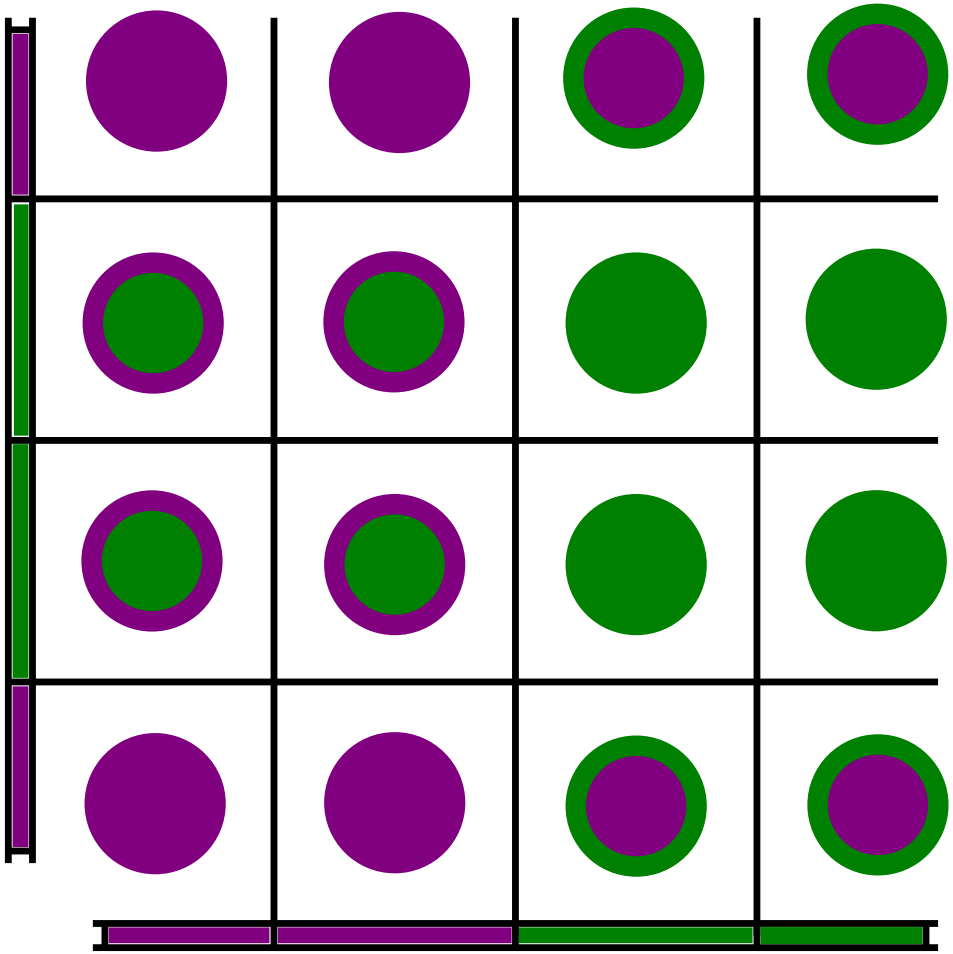
A 2-dimensional example



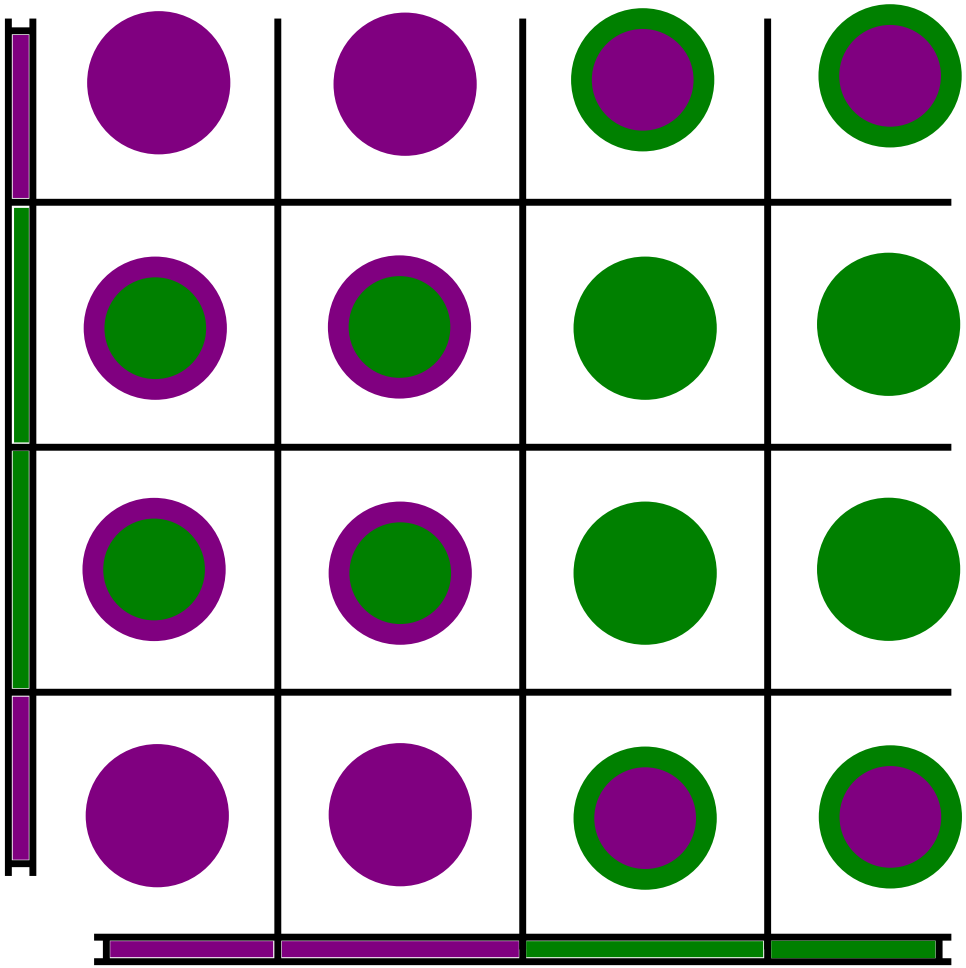
A 2-dimensional example

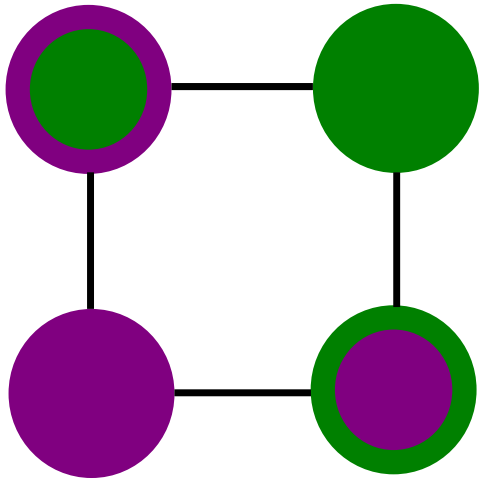


A 2-dimensional example



A 2-dimensional example





The query with Dolphinn

Dolphinn is used for the near neighbors search.

The query with Dolphinn

Dolphinn is used for the near neighbors search.

Thanks to the LSH family, the neighbors of a point tend to be close to it on the hypercube.

The query with Dolphinn

Dolphinn is used for the near neighbors search.

Thanks to the LSH family, the neighbors of a point tend to be close to it on the hypercube.

To find the neighbors of a queried point, its image is computed and the data sharing the same image or a close image are considered as potential neighbors.

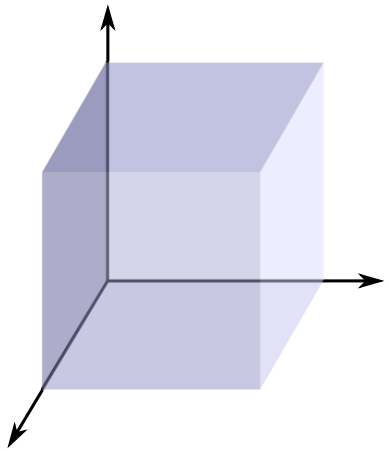
The query with Dolphinn

Dolphinn is used for the near neighbors search.

Thanks to the LSH family, the neighbors of a point tend to be close to it on the hypercube.

To find the neighbors of a queried point, its image is computed and the data sharing the same image or a close image are considered as potential neighbors.

E.g. if 111 is the image of the queried point, the data having for image 111, 110, 101 and 011 are checked.

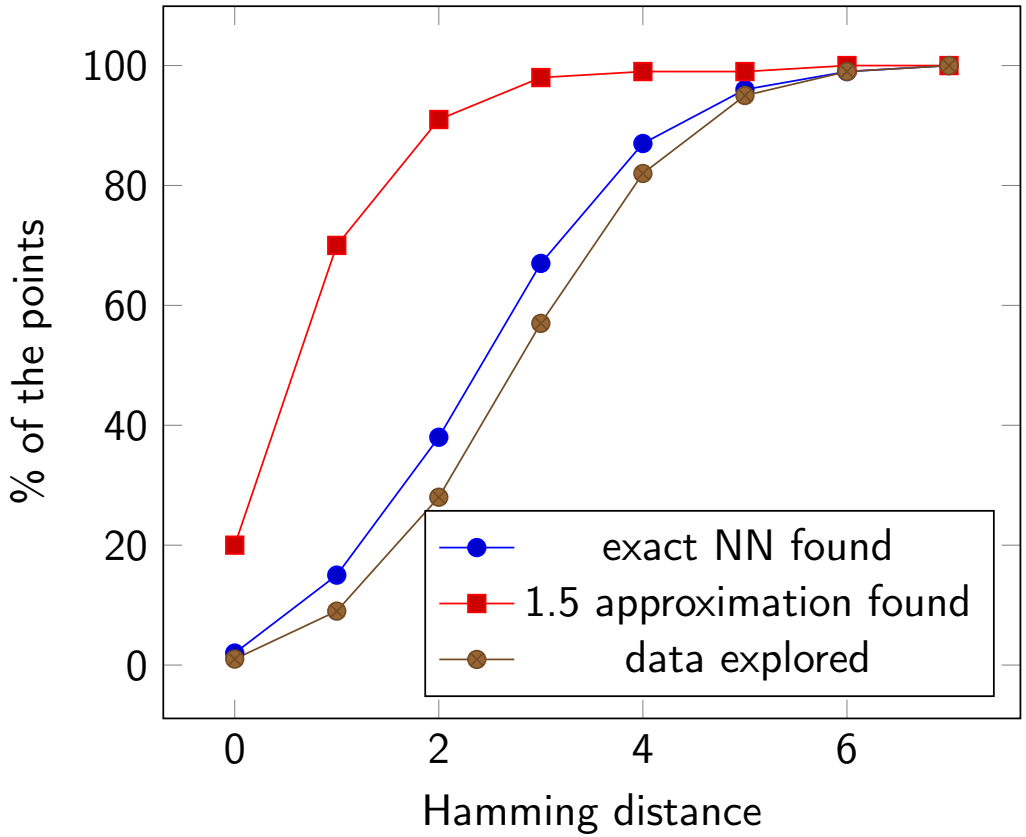


The direct use of Dolphinn

First idea: if Dolphinn allows to find neighbors, maybe it allows to find some nearest neighbors?

The direct use of Dolphinn

First idea: if Dolphinn allows to find neighbors, maybe it allows to find some nearest neighbors?



Using Dolphinn as a k-means initialization

Second idea: try to improve Dolphinn's partition.

Using Dolphinn as a k-means initialization

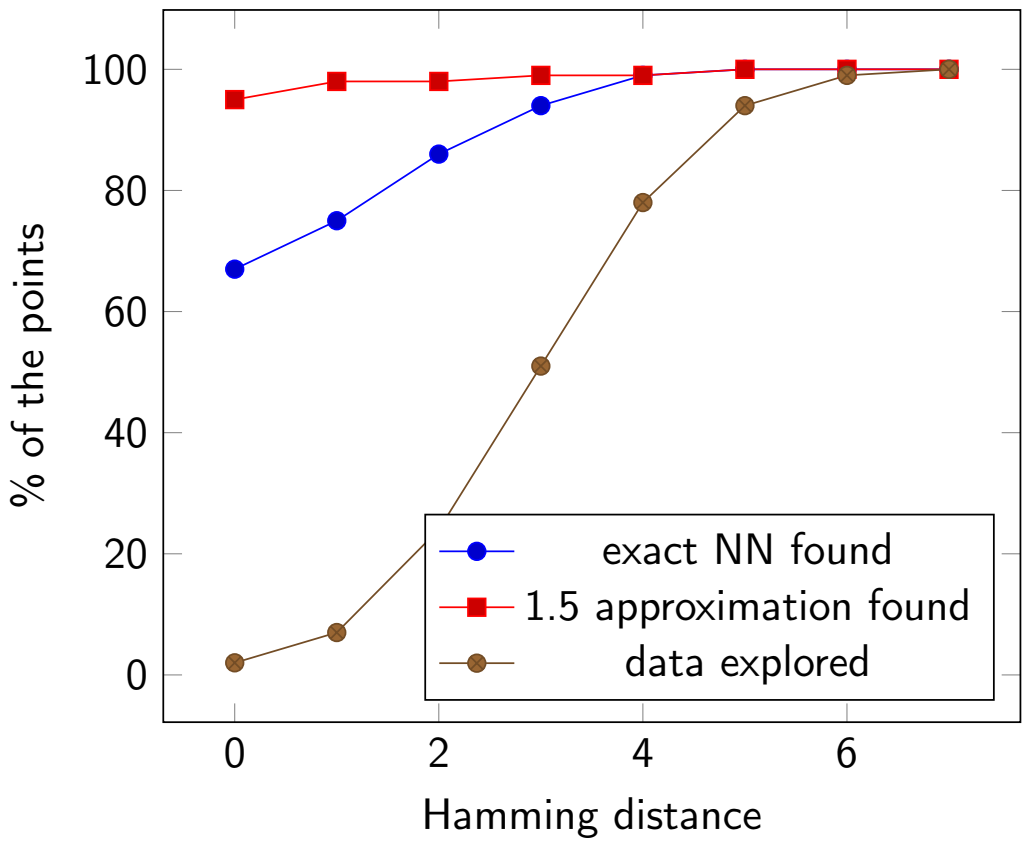
Second idea: try to improve Dolphinn's partition.

- Compute the mean point of every vertex of the hypercube.
- Reassign each point to its closest vertex.

Using Dolphinn as a k-means initialization

Second idea: try to improve Dolphinn's partition.

- Compute the mean point of every vertex of the hypercube.
- Reassign each point to its closest vertex.



The r-nets

Definition

Given a pointset $X \subset \mathbb{R}^d$ and a parameter $r > 0$, an r-net of X is a subset N such that the following properties are met:

- (packing) for every $p \neq q \in N$, $||p - q||_2 > r$
- (covering) for every $p \in X$, there exist $q \in N$ s.t. $||p - q||_2 \leq r$

Using nets to solve the problem

Description of the idea:

- one net for the red and one for the blues;
- try to match one blue cell with one red: the neighbors of the reds in the cell should be close;
- the neighbors of the next red cell should be neighbors of the current blue cell.

Using Dolphinn as a preprocessing to have an idea of the cells of the net.

Using nets to solve the problem

Description of the idea:

- one net for the red and one for the blues;
- try to match one blue cell with one red: the neighbors of the reds in the cell should be close;
- the neighbors of the next red cell should be neighbors of the current blue cell.

Using Dolphinn as a preprocessing to have an idea of the cells of the net.

Some technical problems

- A lot of cells with very few members;
- We obtain a net with a lot of unconnected components

Conclusion

Results

- An empirical method that gives results but no guarantees.
- A lot of technical problems with the nets.

Future work

- Study of different forms of clustering (based or not on the hypercube).
- Comparison to the well separated pair method.
- Focus on more precise distributions rather than on the whole general problem.