On the Complexity of Reverse Similarity Search

Matthew Skala
David R. Cheriton School of Computer Science
University of Waterloo

April 12, 2008
SISAP'08

Outline

- Definitions: metric spaces, VP- and GH-trees, reverse similarity
- Tree metrics
- Hamming and Levenshtein distances
- Superghost distance
- Vectors and L_p metrics
- Conclusion

Metric spaces

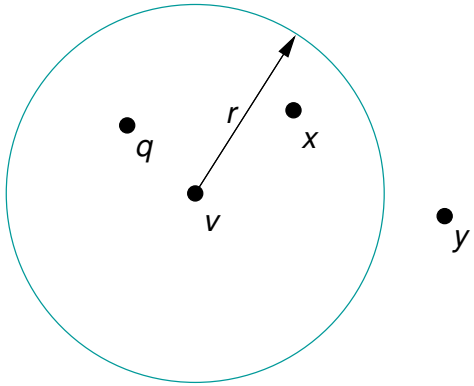
Metric spaces are a general, but rigorous, way of describing any situation where there are things and distances between them.

A metric space is a tuple $\langle S, d \rangle$ of a set S and a function $d : S \times S \to \mathbb{R}$ satisfying $d(x, y) \geq 0$; $d(x, y) = 0$ iff $x = y$; $d(x, y) = d(y, x)$; and the Triangle Inequality $d(x, z) \geq d(x, y) + d(y, z)$.

Many kinds of data can be represented by points in metric spaces; and many kinds of queries can be described in terms of metric spaces.

VP-trees

VP-trees [Yianilos, 1993] are binary trees that split the space at every internal node, based on whether points are within a sphere centred on a Vantage Point (VP) stored in the node. One subtree for "inside," one for "outside."

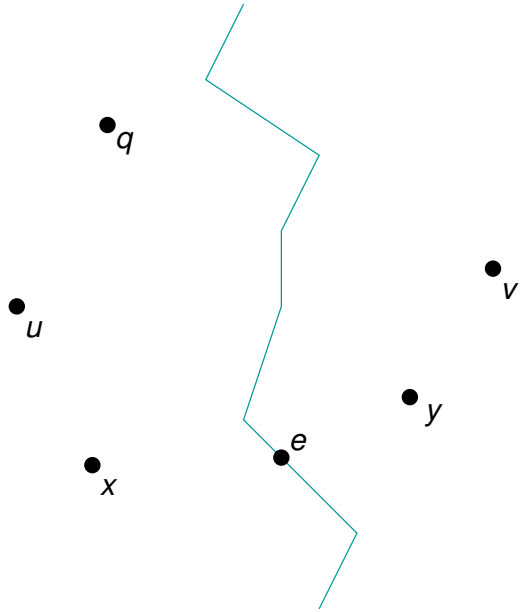


The algorithm to answer a query q descends the tree looking at the vantage points v; it can use the Triangle Inequality to rule points x and y in or out of the result, thereby pruning subtrees from consideration.

GH-trees

GH-trees [Uhlmann, 1991] are binary space partitioning trees too, but use two vantage points per internal node, dividing the space on the Generalized Hyperplane (GH) of points equidistant from them.

VP- and GH-trees are examples of distance based data structures. Note they use no geometric properties of the spaces, only the opaque operation of measuring distance between points.



Reverse similarity search

Guess which city I'm thinking of!

> It's within 1000km of Cancún.
> It's not within 2000km of Rio de Janeiro.
> It's within 3000km of Tokyo.

Alternatively:

> It's closer to Cancún than to Guadalajara.
> It's closer to Rio de Janeiro than to Santiago.
> It's closer to Tokyo than to Beijing.

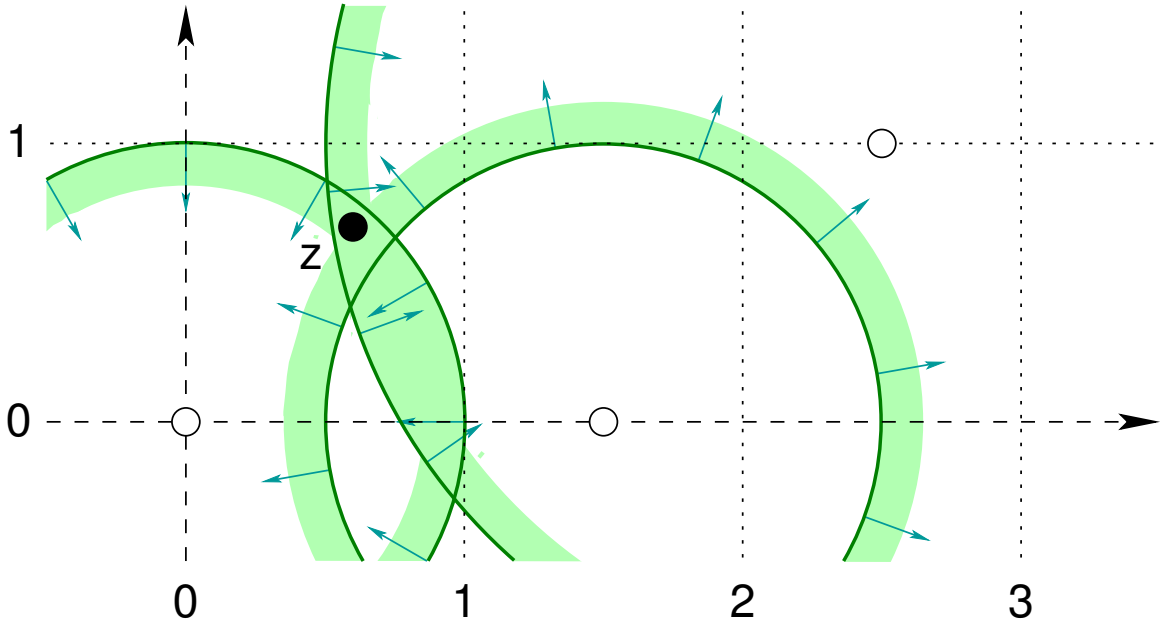
Do such cities exist? How hard is that question?

Application: security of robust hashes.

VPREVERSE

An instance is a set P of ordered triples (x_i, r_i, b_i) with $x_i \in$ the metric space $\langle S, d \rangle$, r_i real, $b_i \in \{0, 1\}$. Accept iff there exists a $z \in S$ such that for every $(x_i, r_i, b_i) \in P$, $d(z, x_i) \leq r_i$ iff $b_i = 1$.

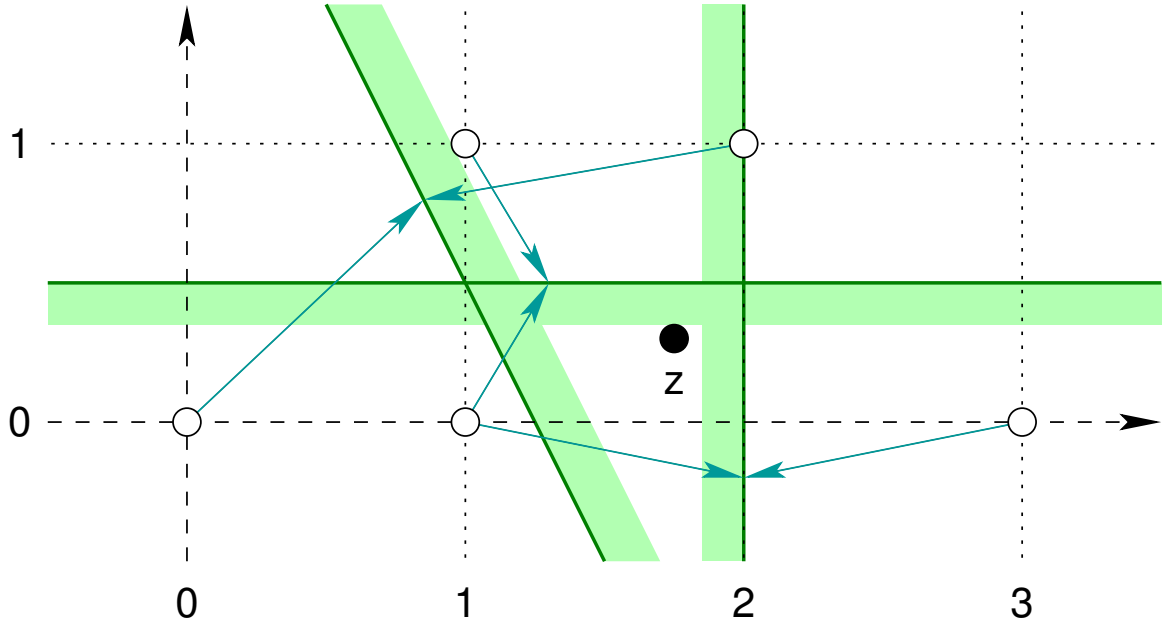
The point z is a poly-time certificate for a "yes" answer.



GHREVERSE

An instance is a set P of ordered pairs of points from $\langle S, d \rangle$. Accept iff there exists a point z such that $d(z, x_i) \leq d(z, y_i)$ for every $(x_i, y_i) \in P$.

As with VPREVERSE, the point z is a poly-time certificate.

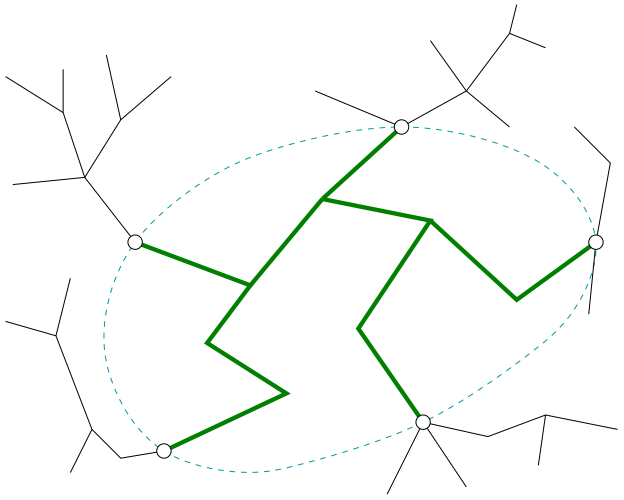


Tree metrics

Points are vertices of a tree, distances are the (possibly weighted) lengths of the unique paths through the tree. *Prefix distance* is one important example: edit a string at one end only. Positive obfuscation results known for tree metrics.

GHREVERSE in $\mathcal{P}$, VPREVERSE usually in $\mathcal{P}$ (but we can construct pathological spaces where it isn't).

Proof concept: Find the spanning subtree of all points mentioned in the instance; that's the place to look for solutions.



Hamming and Levenshtein distances

Hamming edits a string by substitutions anywhere; Levenshtein edits a string with insertions, deletions, or substitutions anywhere.

Reverse similarity search is $\mathcal{NP}$-complete in these spaces. Hamming from work by Frances and Litman [1997] on covering radius, Levenshtein by reduction to Hamming.

Hamming proof concept: double dimensions, so $0 \rightarrow 00$ and $1 \rightarrow 11$, then 01 becomes a don't-care value equidistant to both.

Levenshtein proof concept: add linear amount of padding between all the digits, then the minimal edit sequence has to consist of Hamming moves.

Superghost distance 

If we allow many edits (Hamming, Levenshtein), then reverse similarity search is $\mathcal{NP}$-complete; if we allow very few (prefix), it's in $\mathcal{P}$. I wish it could be both...

Define a new distance: edits allowed at both ends but not in the middle, as in the game of Superghost.

For this distance, our problems are still $\mathcal{NP}$-complete.

The proof encodes 3SAT into a linear number of variable descriptions, each of logarithmic size. We can create this by forcing or forbidding the appearance of log-sized substrings, of which there are only polynomially many. Then a little more technical trickery allows us to force satisfaction of every clause.

Vectors with L_p distance

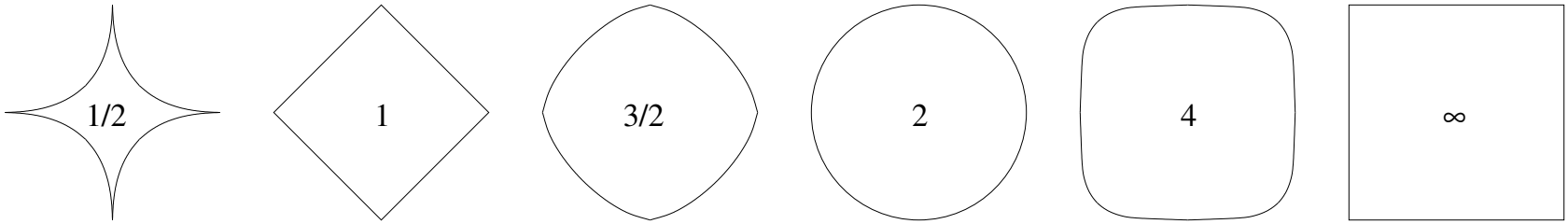
The L_p metrics generalize the Pythagorean Theorem to other exponents:

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

for real $p \geq 1$ (why not $p < 1$?) or

$$d_\infty(\mathbf{x}, \mathbf{y}) = \max_{i=1}^{n} |x_i - y_i|$$

for $p = \infty$.

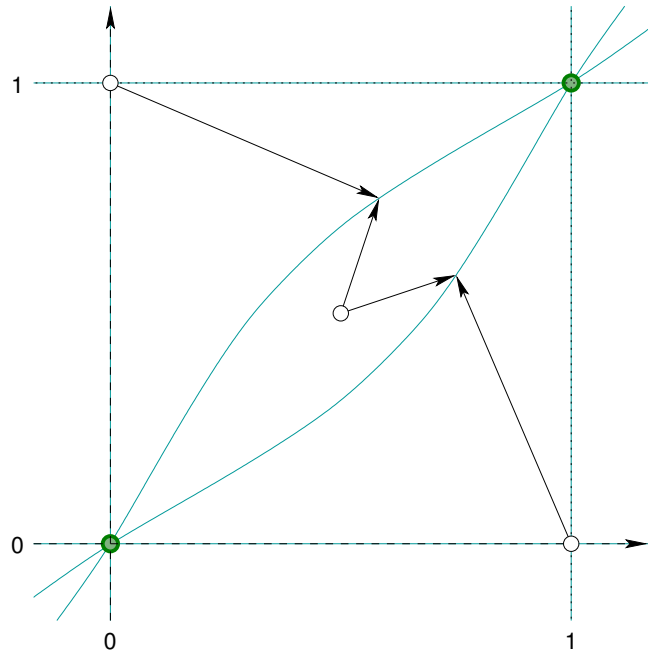


Vector results

GHREVERSE is equivalent to linear programming for L_2 (Euclidean) space. VPREVERSE, and GHREVERSE for other L_p, are $\mathcal{NP}$-complete.

Proof concept: in all cases but that one, we have nonconvex sets to play with. Intersect the nonconvex sets in such a way as to create an exponentially complicated set that encodes an $\mathcal{NP}$-complete problem.

Dimension doubling comes into play for GHREVERSE, much like that seen in the Hamming-strings case.



Conclusions and future work

These problems highlight some interesting properties of different metric spaces, with differences among spaces that otherwise appear similar.

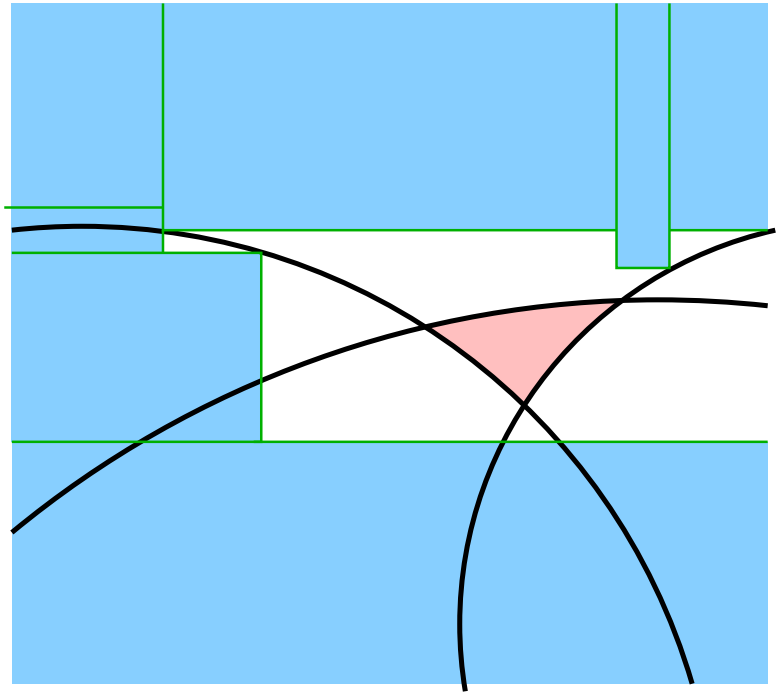
In particular, Euclidean space is very special because of its convex half-spaces, which make GHREVERSE easy.

Superghost distance may have wider application as an in-between kind of metric space.

What about the difficulty and satisfiability of randomly chosen instances?

Extra: How to solve an instance?

Recursively subdivide space into possibly unbounded blocks, and for each one, rule in or out or subdivide. Try to subdivide where it will make future rule-in/out easier.



Extra: A badly-behaved tree space

If $\mathcal{P} \neq \mathcal{UP}$, we can make it easy to measure distances but hard to find paths in a tree space, which breaks the poly-time algorithm for reverse similarity search.