Constraint satisfaction in string spaces

Matthew Skala

University of Waterloo*
University of Toronto†

January 26, 2009

Outline

- Whining and excuses
- Definitions and similarity search
- *Reverse* similarity search
- Results for tree metrics
- Hamming and Levenshtein
- Superghost
- Vectors
- Distance permutations
- Questions answered and not

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.

Vectors with L_p distance

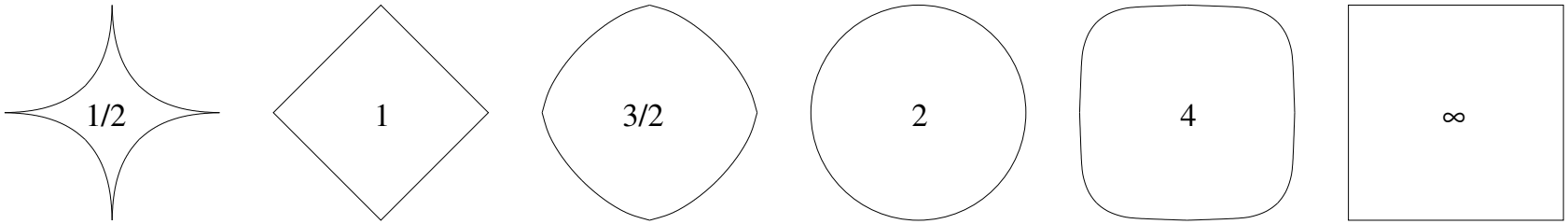
The L_p metrics generalize the Pythagorean Theorem to other exponents:

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

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 $p = \infty$.



Similarity search

Here's a book: 

Here's a library containing $\sim 2 \times 10^6$ books: 

How can I find other books like *Heart of Darkness*?

nearest-neighbour query

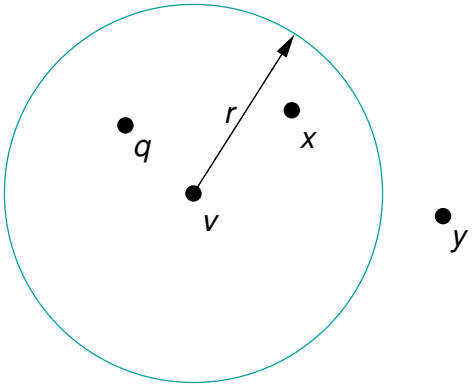
$$\arg\min_{x \in \text{  }} d\left(x, \text{  }\right)$$

range query

$$\left\{x \,\middle|\, d\left(x, \text{  }\right) < r\right\}$$

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."

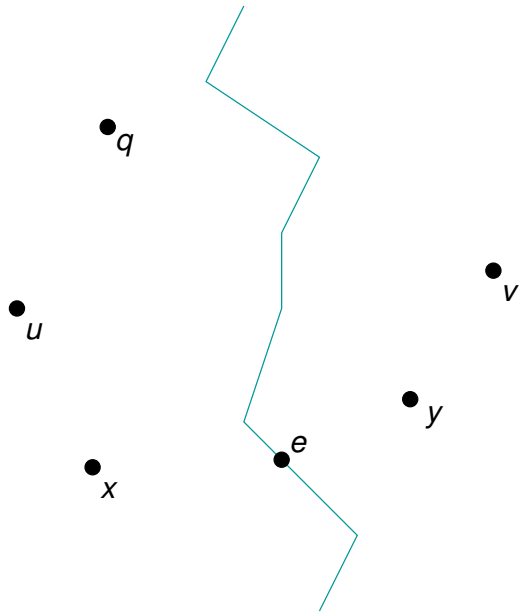


The algorithm to answer a query q descends the tree looking at the vantage points v; it can use the Triangle Inequality to rule points x and y in or out of the result, thereby pruning subtrees from consideration.

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.

Robust hashing

> "I shall not today attempt further to define the kinds of material I understand to be embraced within that shorthand description; and perhaps I could never succeed in intelligibly doing so. But I know it when I see it, and the motion picture involved in this case is not that." —Potter Stewart

The problem: write a shorthand description to enable Justice Stewart to identify the motion picture when he sees it, without actually showing him an example.

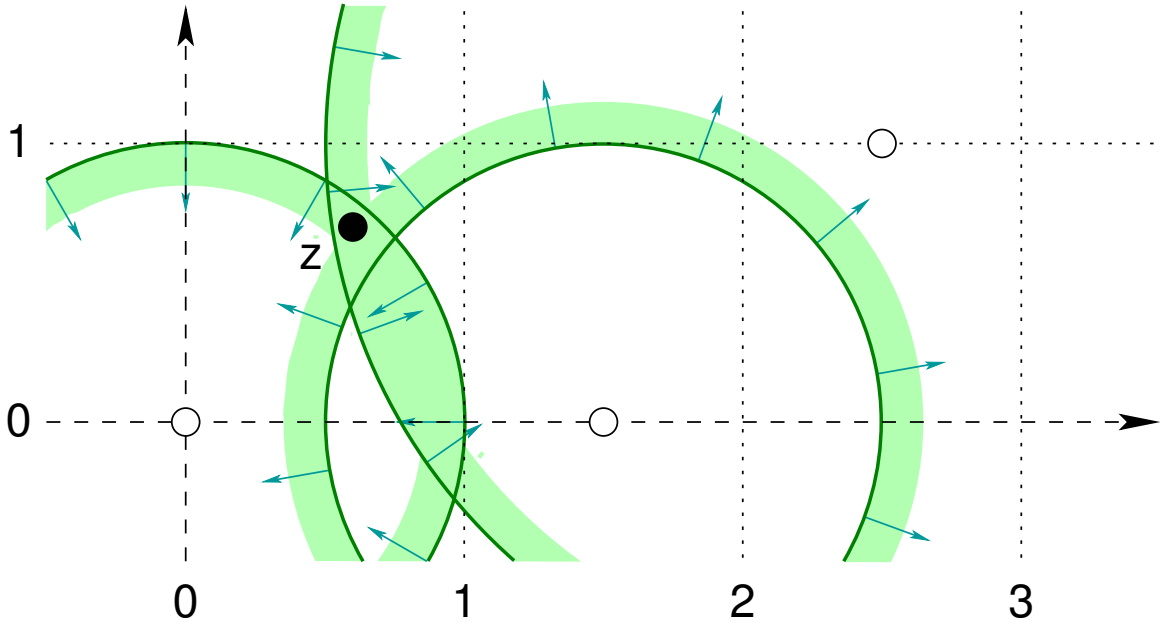
For exact match: the crypto people consider it a solved problem, even though Wang, Yin, and Yu have broken all current techniques.

For fuzzy match: we want some kind of secure sketch.

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 $b_i = 1$.

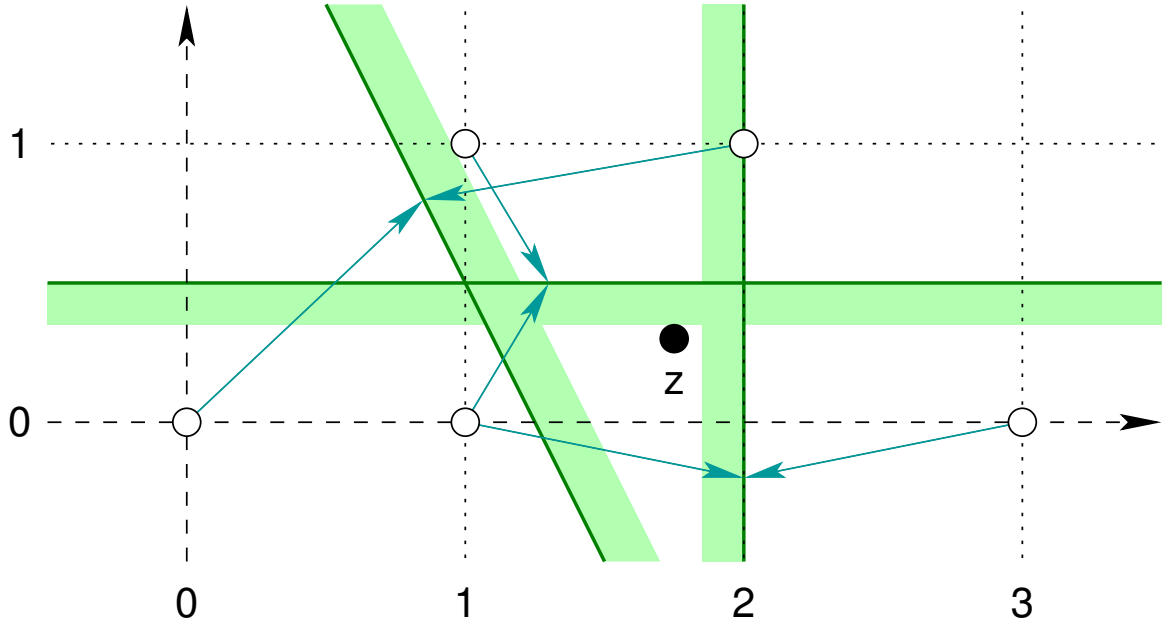
The point z is a poly-time certificate for a "yes" answer.



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.

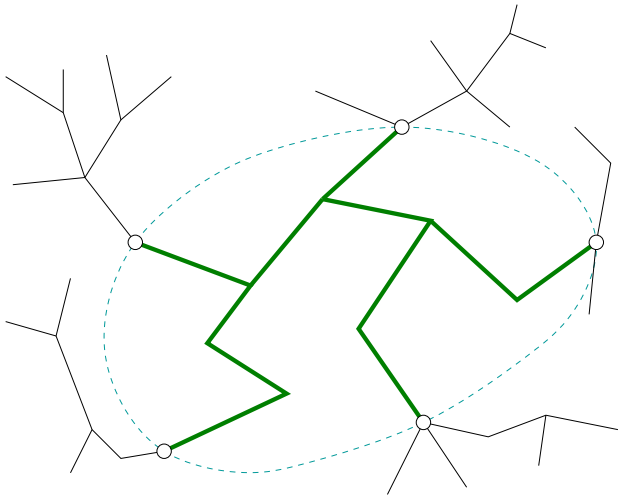


Tree metrics

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.



Label all the books

Call number prefixes map differences among books into a tree metric:

- "P": language and literature
- "PR": English literature
- "PR60...": 20th Century English authors
- "PR6005": before 1950, surname starts with C...
- "PR6005.O": surname starts with CO...
- "PR6005.O4": Joseph Conrad
- "PR6005.O4H4": *Heart of Darkness*

Linear VPREVERSE on finite explicit trees

(What's an explicit tree?)

Algorithm concept: When we sit on an edge looking towards one vertex, all that matters about the part of the tree behind us is the *interval* of radii permitted by constraints centred behind us. That's a small amount of information.

By doing two depth-first searches, in linear time, we can propagate all the constraints all over the tree and find any 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.

Ghost example

Alice:	I	
Bob:	IN	
Alice:	INS	
Bob:	INSO	(hoping for INSOFAR)
Alice:	INSOL	(hoping for INSOLENT)
Bob:	INSOLU	
Alice:	INSOLUB	(not much choice here)
Bob:	INSOLUBL	
Alice:	INSOLUBLE	(she becomes $\frac{1}{3}$ of a ghost)

Superghost

We take turns adding letters to either end of a string. At all times it must remain a substring of a dictionary word; however, if it becomes equal to a dictionary word then the player who added the last letter loses.

> "Starting words in the middle and spelling them in both directions lifts the pallid pastime of Ghosts out of the realm of childrens parties and ladies sewing circles and makes it a game to test the mettle of the mature adult mind."
> —James Thurber

My hope: some bioinformatics person will say "Oh, yeah, that's just like such-and-such polymerization that we're studying!"

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.

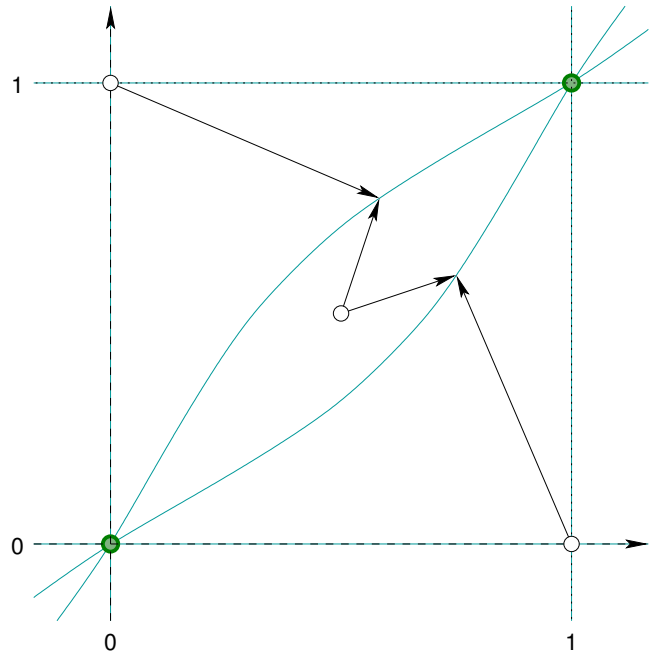
The proof encodes 3SAT into a linear number of variable descriptions, each of logarithmic size. We can create this by forcing or forbidding the appearance of log-sized substrings, of which there are only polynomially many. Then a little more technical trickery allows us to force satisfaction of every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

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.



Distance Permutations [Chávez, Figueroa, and Navarro, 2005]

I have a fixed list of 10 books everybody knows.

I evaluate the distance from *Heart of Darkness* to each of the books on the list.

If I wrote down that 10-vector to use with the triangle inequality, I'd be doing LAESA. [Micó and Vidal, 1994]

But instead, I'll just sort it and write down the resulting permutation. That will be my label I put on the book.

Similar books ought to have similar labels.

Labelling *Heart of Darkness*

r_i	i	Title
1	6	*Life of Pi*
2	7	*Ragtime*
3	0	*Historia universal de la infamia*
4	2	*The Snarkout Boys and the Avocado of Death*
5	5	*El ingenioso hidalgo don Quijote de la Mancha*
6	1	*2001, a Space Odyssey*
7	8	*Teatro herético*
8	3	*The Art of Computer Programming*
9	9	*Good to Great*
10	4	*LaTeX: a Document Preparation System*

The label for *Heart of Darkness* is "6702518394". How many distinct labels are possible?

Questions answered and not

These kinds of constraint satisfaction tend to be hard on all but the simplest string spaces.

Does Superghost distance have any interesting applications?

Any nontrivial string metrics with less than $\mathcal{NP}$-complete behaviour?

The original robust hashing/secure sketching questions; these results actually argue against security, because of Deep Magic involving the polynomial hierarchy.

Distance permutation counting in string spaces?

Extra bonus slide!

Suppose I want you to be able to search for a substring without knowing the substring until you see it. (Application: "Cypher Patrol" Web censorship.) Blind Substring Search [Skala 1998].

Possible solution: hash text in such a way that each bit of M' depends on a sliding window of k bits surrounding the same position in M. If k significantly less than the substring size, the hashed substring should occur iff the original substring occurs. But you have to supply the missing bits in order to reverse the hash.

In practice, not very good. Is anything better possible?