ON THE APPROXIMABILITY OF NUMERICAL TAXONOMY (FITTING DISTANCES BY TREE METRICS)

RICHA AGARWALA, VINEET BAFNA, MARTIN FARACH, MIKE PATERSON, AND MIKKEL THORUP

October 30, 2018

Overview

What is an approximation algorithm?

Definition 1.1

An algorithm $\mathcal{A}$ for an optimization problem X is an $a-$approximation algorithm iff

- For each instance I of X, the algorithm $\mathcal{A}$ outputs a valid solution I
- $\mathcal{A}$ is a polynomial-time algorithm
- If $\mathrm{OPT}(I)$ is the value of the optimum solution to I and $\mathcal{A}(I)$ is the value of solution for I output by $\mathcal{A}$, then

$$\frac{\mathrm{OPT}(I)}{\mathcal{A}} \leq a$$

$$\frac{\mathcal{A}}{\mathrm{OPT}(I)} \leq a$$

The numerical taxonomy problem

Given a distance matrix $D : S^2 \to \mathcal{R}_{\geq 0}$, produce a *tree metric* T which spans S and fits D.

- In the general case, we want an *additive* tree metric.
- By “fitting,” we mean producing a T such that $\|D - T\|_k$ for some k.

The numerical taxonomy problem

Given a distance matrix $D : S^2 \to \mathcal{R}_{\geq 0}$, produce a *tree metric* T which spans S and fits D.

- In the general case, we want an *additive* tree metric.
- By "fitting," we mean producing a T such that $\|D - T\|_k$ for some k.

Remark 1

If there exists T such that the error is 0, T is unique and constructible in poly-time.

Remark 2

It has been shown that for L_1 and L_2 norms, the problem is NP*-Hard, even for the ultrametric case.*

Importance and contributions

Being able to provide additive metrics given any metric is an extremely generic and popular task. It is easy to imagine cases where you have a similarity matrix and would like a tree from it. This paper makes 2 main contributions to the problem. While both contributions are fairly theoretical as opposed to application-based, they help pave the way for better approximations.

- 3-approximation for Numerical Taxonomy
- $\frac{9}{8}$ approximation hardness for Numerical taxonomy

Importance and contributions

Being able to provide additive metrics given any metric is an extremely generic and popular task. It is easy to imagine cases where you have a similarity matrix and would like a tree from it. This paper makes 2 main contributions to the problem. While both contributions are fairly theoretical as opposed to application-based, they help pave the way for better approximations.

- 3-approximation for Numerical Taxonomy
- $\frac{9}{8}$ approximation hardness for Numerical taxonomy

Remark 3

There is a lot of room for improvement on the approximation results!

Metrics

Definition 2.1 (4 point condition)

A (quasi) metric D is (quasi) additive if $\forall\ a, b, c, d$

$$D[a,b] + D[c,d] \leq \max\{D[a,c] + D[b,d], D[a,d] + D[b,c]\}$$

Metrics

Definition 2.1 (4 point condition)

A (quasi) metric D is (quasi) additive if $\forall\ a, b, c, d$

$$D[a, b] + D[c, d] \leq \max\{D[a, c] + D[b, d], D[a, d] + D[b, c]\}$$

Theorem 2.2

A metric is additive iff it is a tree metric.

Metrics

Definition 2.1 (4 point condition)

A (quasi) metric D is (quasi) additive if $\forall\ a, b, c, d$

$$D[a,b] + D[c,d] \leq \max\{D[a,c] + D[b,d], D[a,d] + D[b,c]\}$$

Theorem 2.2

A metric is additive iff it is a tree metric.

Definition 2.3

A metric D is an ultrametric if $\forall\ a, b, c$

$$D[a,b] \leq \max\{D[a,c], D[b,c]\}$$

Metrics

Definition 2.4

A quasimetric D on n objects is a centroid quasimetric if $\exists \ell_1, \ldots, \ell_n$ such that $\forall i \neq j, D[i,j] = \ell_i + \ell_j$

Remark 4

This is a star tree.

Metrics

Definition 2.4

A quasimetric D on n objects is a centroid quasimetric if $\exists \ell_1, \ldots, \ell_n$ such that $\forall i \neq j, D[i,j] = \ell_i + \ell_j$

Remark 4

This is a star tree.

Definition 2.5

$$\|M\|_\infty = \max_{i<j}\{|M[i,j]|\}$$

Definitions and Lemmas

We need some base machinery to get started.

Definitions and Lemmas

We need some base machinery to get started.

- $m_a = \max_i\{D[a,i]\}$.

Definitions and Lemmas

We need some base machinery to get started.

- $m_a = \max_i\{D[a,i]\}$.
- C^a be the centroid metric with $\ell_i = m_a - D[a,i]$

Definitions and Lemmas

We need some base machinery to get started.

- $m_a = \max_i\{D[a,i]\}$.
- C^a be the centroid metric with $\ell_i = m_a - D[a,i]$

Lemma 3.1

For any point a, D is quasiadditive if and only if $D + C^a$ is an ultrametric.

Definitions and Lemmas

We need some base machinery to get started.

- $m_a = \max_i\{D[a, i]\}$.
- C^a be the centroid metric with $\ell_i = m_a - D[a, i]$

Lemma 3.1

For any point a, D is quasiadditive if and only if $D + C^a$ is an ultrametric.

Lemma 3.2

Given an additive metric A and a centroid quasi-metric Q,

$$A + Q \text{ is additive} \iff A + Q \text{ satisfies the triangle inequality}$$

Notation

Let $\mathcal{X}$ be all additive metrics and $\mathcal{X}^a$ be all a-restricted additive metrics.

Definition 3.3

Metric M is a-restricted if $\forall i, M[a,i] = D[a,i]$.

Notation

Let $\mathcal{X}$ be all additive metrics and $\mathcal{X}^a$ be all a-restricted additive metrics.

Definition 3.3

Metric M is a-restricted if $\forall i, M[a,i] = D[a,i]$.

We define $\mathcal{A}(D)$ and $\mathcal{A}^a(D)$ to be additive metrics on D such that

Definition 3.4

$$\|D - \mathcal{A}(D)\|_\infty = \min_{A \in \mathcal{X}} \|D - A\|_\infty$$

$$\|D - \mathcal{A}^a(D)\|_\infty = \min_{A \in \mathcal{X}^a} \|D - A\|_\infty$$

Notation

Let $\mathcal{X}$ be all additive metrics and $\mathcal{X}^a$ be all a-restricted additive metrics.

Definition 3.3

Metric M is a-restricted if $\forall i, M[a,i] = D[a,i]$.

We define $\mathcal{A}(D)$ and $\mathcal{A}^a(D)$ to be additive metrics on D such that

Definition 3.4

$$\|D - \mathcal{A}(D)\|_\infty = \min_{A \in \mathcal{X}} \|D - A\|_\infty$$

$$\|D - \mathcal{A}^a(D)\|_\infty = \min_{A \in \mathcal{X}^a} \|D - A\|_\infty$$

Similarly for $\mathcal{U}(D)$ and $\mathcal{U}^a(D)$ but on ultrametrics.

Remark 5

$\mathcal{U}$ is computable in $O(n^2)$ time.

Inspiration

Armed with Lemmas 3.1 and 3.2, we know that if we take $\mathcal{U}(D + C^a)$, which is the closest ultrametric on $D + C^a$, then we can remove the C^a addition from the optimal ultrametric and obtain some quasi-additive approximation D' of D i.e.

$$D' = \mathcal{U}(D + C^a) - C^a$$

. By Lemma 3.2, we know D' is only additive if it satisfies the triangle inequality.

Inspiration

Armed with Lemmas 3.1 and 3.2, we know that if we take $\mathcal{U}(D + C^a)$, which is the closest ultrametric on $D + C^a$, then we can remove the C^a addition from the optimal ultrametric and obtain some quasi-additive approximation D' of D i.e.

$$D' = \mathcal{U}(D + C^a) - C^a$$

. By Lemma 3.2, we know D' is only additive if it satisfies the triangle inequality.
We also know from Lemma 3.1, that $\mathcal{A}^a(D) + C^a$ is an ultrametric. If we can also show that it is a-restricted, then clearly $\mathcal{U}^a(D + C^a)$ is at least a good approximation of $D + C^a$ as $\mathcal{A}^a(D) + C^a$.

Ultrametric algorithm

Theorem 3.5

Given two distance metrics L, M and some value h such that $L[i,j] \leq M[i,j] \leq h$, we can compute an optimal ultrametric U for M that satisfies $L[i,j] \leq U[i,j] \leq h$ in $O(n^2)$ time.

Algorithm 1 Optimal Ultrametric

```
function ULTRA(T, w)
    T' = MST(T)
    e_ij = max_{e ∈ T'} w(e)
    T_1, T_2 = PARTITION(T', e_ij)
    T^u = r
    u_1 = ULTRA(T_1, w)
    u_2 = ULTRA(T_2, w)
    Add edges (r, u_1) and (r, u_2) to T^u
    Set w((r, u_1)) = w((r, u_2)) = D[i,j]/2 − HEIGHT(T_1)
    return T^u
end function
```

3-Approximation algorithm

Lemma 3.6

$\forall a, \|\mathcal{A}^a(D) - D\|_\infty \leq 3\,\|\mathcal{A}(D) - D\|_\infty.$

Proof 1

Short proof in paper

Polynomial time algorithm

Lemma 3.7

$\forall a, \mathcal{A}^a(D)$ *can be computed in polynomial time.*

Definition 3.8

Ultrametric U is a restricted iff

- $2m_a \geq U[i,j] \geq 2\max\{\ell_i, \ell_j\}$
- $U[a,i] = 2m_a, i \neq a$

Lemma 3.9

$\mathcal{U}^a(D + C^a) - C^a$ *is an a-restricted additive metric.*

Lemma 3.10

$\mathcal{A}^a(D) + C^a$ *is an a-restricted ultrametric.*

Final proof

Proof 2

$T = \mathcal{U}^a(D + C^a) - C^a$

$$
\begin{aligned}
\|T - D\|_\infty &\geq \|\mathcal{A}^a(D) - D\|_\infty \text{ By claim A} && (1)\\
&= \|\mathcal{A}^a(D) + C^a - (D + C^a)\|_\infty && (2)\\
&\geq \|\mathcal{U}^a(D + C^a) - C^a) - D\|_\infty \text{ By claim B} && (3)\\
&= \|T - D\|_\infty && (4)
\end{aligned}
$$

Final proof

Proof 2

$T = \mathcal{U}^a(D + C^a) - C^a$

$$
\begin{aligned}
\|T - D\|_\infty &\geq \|\mathcal{A}^a(D) - D\|_\infty \text{ By claim } A && (1)\\
&= \|\mathcal{A}^a(D) + C^a - (D + C^a)\|_\infty && (2)\\
&\geq \|\mathcal{U}^a(D + C^a) - C^a) - D\|_\infty \text{ By claim } B && (3)\\
&= \|T - D\|_\infty && (4)
\end{aligned}
$$

Theorem 6

$\|T - D\|_\infty = \|\mathcal{A}^a(D) - D\|_\infty \leq 3\,\|\mathcal{A}(D) - D\|_\infty$

What is tightness?

Just like in runtime analysis, our approximation factor a is tight if for some N, there exists a problem instance of size $n > N$ such that our algorithm obtains a value exactly a times the optimal value.

A simple instance

Theorem 7

There is an nxn distance matrix D such that for all c

$$\frac{\|D - \mathcal{A}^c(D)\|_\infty}{\|D - \mathcal{A}(D)\|_\infty} = 3$$

Proof 3

$$\begin{aligned} D[i,j] =& d - \epsilon \text{ if } i = (j+1) \mod 9 \text{ or } i = (j-1) \mod 9 \\ &= 0 \text{ if } i = j \mod 3 \\ &= d + \epsilon \text{ otherwise} \end{aligned}$$

A simple instance

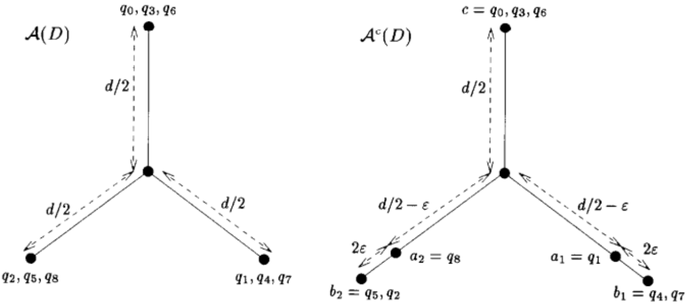

FIG. 4.1. *Trees approximating D.*

Hardness of approximation

Some problems are considered to be harder to approximate than others. This is because, for many NP problems, we can show that there exists no polynomial time algorithm to approximate a solution within some value for that problem.

How hard is the numerical phylogeny approximation problem?

Theorem 8

The decision version of the numerical taxonomy problem is NP-complete.

Proof 4

Reduction from 3SAT

Theorem 9

Finding T such that $\|T - D\|_\infty \leq \frac{9}{8}\mathrm{OPT}$ is NP-hard.

How to show a problem can't be approximated to some factor

The lower bound proof in the problem is a bit more difficult than the previous proofs in the paper. Instead, we will briefly discuss how to prove a lower bound.

How to show a problem can't be approximated to some factor

The lower bound proof in the problem is a bit more difficult than the previous proofs in the paper. Instead, we will briefly discuss how to prove a lower bound. Say we have some problem which we know is NP complete. If we can reduce this problem to an instance of our problem such that there is a gap in thepossible values that a solution to our problem can have. In other words, we reduce some other problem to our problem such that we answer YES if the solution to our current problem is less than c and NO if the solution is greater than $c\alpha$ for some alpha.

How to show a problem can't be approximated to some factor

The lower bound proof in the problem is a bit more difficult than the previous proofs in the paper. Instead, we will briefly discuss how to prove a lower bound. Say we have some problem which we know is NP complete. If we can reduce this problem to an instance of our problem such that there is a gap in thepossible values that a solution to our problem can have. In other words, we reduce some other problem to our problem such that we answer YES if the solution to our current problem is less than c and NO if the solution is greater than $c\alpha$ for some alpha.

If you design the reduction correctly, you can show that if you are able to approximate your problem with some approximation factor, then you can use it to solve a NP problem, which is impossible.

Generalizability

The algebra from the beginning of section 3 can be easily applied to any of the other norms to prove the 3 approximation factor holds.

Questions?

This slide is reserved for questions.