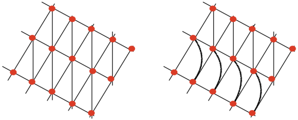
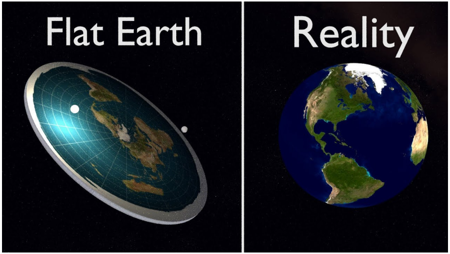
Metric representations: Algorithms and Geometry

Anna C. Gilbert
Department of Mathematics
Department of Statistics & Data Science
Yale University

Joint work with Rishi Sonthalia (University of Michigan)

Distorted geometry or broken metrics





Metric failures

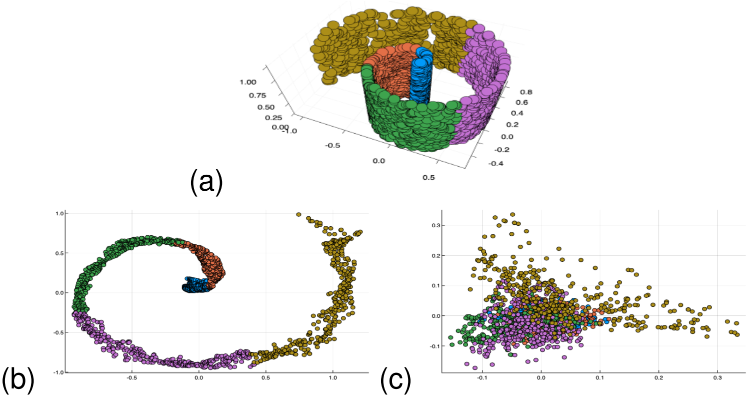
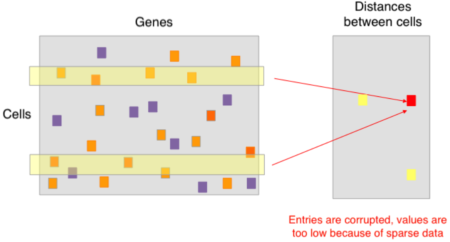


Figure: (a) 2000 data points in the Swissroll. For (b) and (c) we took the pairwise distance matrix and added $2\mathcal{N}(0,1)$ noise to 5% of the distances. We then constructed the 30-nearest-neighbor graph G from these distances, where roughly 8.5% of the edge weights of G were perturbed. For (b) we used the true distances on G as the input to ISOMAP. For (c) we used the perturbed distances.

Motivation



Performance of many ML algorithms depends on the quality of metric representation of data.

Metric should capture salient features of data.

Trade-offs in capturing features and exploiting specific geometry of space in which we represent data.

Representative problems in metric learning

Metric nearness: given a set of distances, find the closest (in ℓ_p norm, $1 \leq p \leq \infty$) metric to distances

Correlation clustering: partition nodes in graph according to their similarity

Metric learning: learn a metric that is consistent with (dis)similarity information about the data

Definitions

$d =$ distance function $X \to \mathbb{R}$

$D =$ matrix of pairwise distances

$G = (V, E, w) =$ graph induced by data set X

$\mathrm{MET}_n =$ metric polytope

$\mathrm{MET}_n(G) =$ projection of MET_n onto coordinates given by edges E of G

Observation: $x \in \mathrm{MET}_n(G)$ iff $\forall e \in E, x(e) \geq 0$ and for every cycle C in G and all $e' \in C$,

$$x(e) \leq \sum_{e' \in C, e' \neq e} x(e');$$

i.e., $\mathrm{MET}_n(G)$ is the intersection of (exponentially many) half spaces.

Specific problem formulations

Correlation clustering

Given graph G and (dis)similarity measures on each edge e, $w^+(e)$ and $w^-(e)$, partition nodes into clusters a la

$$\min \sum_{e \in E} w^+(e)x_e + w^-(e)(1 - x_e) \quad \text{where } x_e \in \{0, 1\}, \text{ or}$$

$$\min \sum_{e \in E} w^+(e)x_e + w^-(e)(1 - x_e) \quad \text{s.t. } x_{ij} \le x_{ik} + x_{kj},\ x_{ij} \in [0, 1].$$

Metric nearness

Given D, $n \times n$ matrix of distances, find closest metric

$$\hat{M} = \arg\min \|D - M\|_p \quad \text{s.t.} \quad M \in \mathrm{MET}_n.$$

Tree and $\delta-$hyperbolic metrics

$$\hat{T} = \arg\min \|D - T\|_2 \quad \text{s.t. } T \text{ is a tree.}$$

Specific problem formulations, cont'd

General metric learning
Given $\mathcal{S} = \{(x_i, x_j)\}$ similar pairs and $\mathcal{D} = \{(x_k, x_l)\}$ dissimilar pairs, we seek a metric $\hat{M}$ that has small distances between pairs in $\mathcal{S}$ and large between those in $\mathcal{D}$

$$\hat{M} = \arg\min \lambda \sum_{(x,x') \in \mathcal{S}} M(x, x') - (1 - \lambda) \sum_{(x,x') \in \mathcal{D}} M(x, x')$$

$$\text{s.t. } M \in \mathrm{MET}_n.$$

General problem formulation: metric constrained problems

Given a strictly convex function f, a graph G, and a finite family of half-spaces $\mathcal{H} = \{H_i\}$, $H_i = \{x \mid \langle a_i, x \rangle \leq b_i\}$, we seek the unique point $x^* \in \bigcap_i H_i \bigcap \mathrm{MET}_n(G)$ that minimizes f

$$x^* = \arg\min f(x) \quad \text{s.t. } Ax \leq b,\ x \in \mathrm{MET}_n(G).$$

Note: A encodes additional constraints such as $x_{ij} \in [0, 1]$ for correlation clustering, e.g.

Optimization techniques: existing methods

Constrained optimization problems with **many** constraints: $O(n^3)$ for simple triangle inequality constraints, possibly exponentially many for graph cycle constraints.

Existing methods don't scale

- too many constraints
- stochastic sampling constraints: too many iterations
- Lagrangian formulations don't help with scaling or convergence problems

Project and Forget

Iterative algorithm for convex optimization subject to metric constraints (possibly exponentially many)

Project: Bregman projection based algorithm that does not need to look at the constraints cyclically

Forget: constraints for which we haven't done any updates

Algorithm converges to the global optimal solution, optimality error decays exponentially asymptotically

When algorithm terminates, the set of constraints are exactly the active constraints

Stochastic variant

Project and Forget

Algorithm 1 General Algorithm.

function F(f)
    $L^{(0)} = \emptyset$, $z^{(0)} = 0$. Initialize $x^{(0)}$ so that $\nabla f(x^{(0)}) = 0$.
    while Not Converged **do**
        $L = \text{METRIC VIOLATIONS}(x^{\nu})$
        $\tilde{L}^{(\nu+1)} = L^{(\nu)} \cup L \cup \mathscr{A}$
        $x^{(\nu+1)} = \text{Project}(x^{(\nu)}, \tilde{L}^{(\nu+1)})$
        $L^{(\nu+1)} = \text{Forget}(\tilde{L}^{(\nu+1)})$
    return x

Algorithm 2 Project and Forget algorithms.

function PROJECT(x, z, L)
    for $H_i = \{y : \langle a_i, y\rangle = b_i\} \in L$ **do**
        Find x^*, θ by solving $\nabla f(x^*) - \nabla f(x) = \theta a_i$ and $x^* \in H_i$
        $c_i = \min(z_i, \theta)$
        $x \leftarrow$ such that $\nabla f(x^{n+1}) - \nabla f(x) = c_i a_i$
        $z_i \leftarrow z_i - c_i$
    return x, z

function FORGET(x, z, L)
    for $H_i = \{x : \langle a_i, x\rangle = b_i\} \in L$ **do**
        if $z_i == 0$ **then** Forget H_i
    return L

Metric violations: Separation oracle

Constraints may be so numerous, writing them down is computationally infeasible. Access them only through a separation oracle.

Property 1: $\mathcal{Q}$ is a deterministic separation oracle for a family of half spaces $\mathcal{H}$ if there exists a positive, non-decreasing, continuous function φ (with $\varphi(0) = 0$) such that on input $x \in \mathbb{R}^d$, $\mathcal{Q}$ either certifies $x \in C$ or returns a list $L \subset \mathcal{H}$ such that

$$\max_{C' \in L} \operatorname{dist}(x, C') \geq \varphi\Big(\operatorname{dist}(x, C)\Big).$$

Stochastic variant: random separation oracle

Metric violations: shortest path

Algorithm 2 Finding Metric Violations.

```
function METRIC VIOLATIONS(()d)
    L = ∅
    Let d(i, j) be the weight of shortest path between nodes i and j or ∞ if none exists.
    for Edge e = (i, j) ∈ E do
        if w(i, j) > d(i, j) then
            Let P be the shortest path between i and j
            Add C = P ∪ {(i, j)} to L
    return L
```

Proposition

METRIC VIOLATION *is an oracle that has Property 1 that runs in* $\Theta(n^2 \log(n) + n|E|)$ *time.*

Bregman projection

Generalized Bregman distance: for a convex function f with gradient $D_f : S \times S \to \mathbb{R}$

$$D_f(x, y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle.$$

Bregman projection: of point y onto closed convex C with respect to D_f is the point x^*

$$x^* = \underset{x \in C \cap \mathrm{dom}(f)}{\arg\min} \; D_f(x, y)$$

Theoretical results: Summary

Theorem

If $f \in \mathcal{B}(S)$, H_i are strongly zone consistent with respect to f, and $\exists\, x^0 \in S$ such that $\nabla f(x^0) = 0$, then

> *Then any sequence x^n produced by Algorithm converges to the optimal solution of problem.*
>
> *If x^* is the optimal solution, f is twice differentiable at x^*, and the Hessian $H := Hf(x^*)$ is positive semidefinite, then there exists $\rho \in (0, 1)$ such that*
>
> $$\lim_{\nu \to \infty} \frac{\|x^* - x^{\nu+1}\|_H}{\|x^* - x^{\nu}\|_H} \leq \rho \tag{0.1}$$
>
> *where $\|y\|_H^2 = y^T H y$.*

The proof of Theorem 2 also establishes another important theoretical property: If a_i is an inactive constraint, then $z_i^{\nu} = 0$ for the tail of the sequence.

Experiments: Weighted correlation clustering (dense graphs)

Veldt, et al. show standard solvers (e.g., Gurobi) run out of memory with $n \approx 4000$ on a 100 GB machine.

Veldt, et al. develop a method for $n \approx 11000$, transform problem to

$$\begin{array}{ll} \text{minimize} & \tilde{w}^T |x - d| + \frac{1}{\gamma} |x - d|^T W |x - d| \\ \text{subject to} & x \in \text{MET}(K_n) \end{array}$$

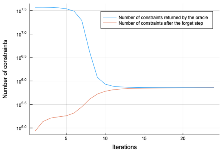
We solve this version of the LP, compare on 4 graphs from the Stanford network repository in terms of running time, quality of the solutions, and memory usage.

Experiments: Weighted correlation clustering (dense graphs)

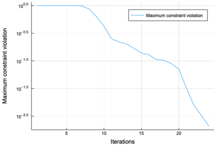
Table 1: Table comparing PROJECT AND FORGET against Ruggles et al. [25] in terms of time taken, quality of solution, and average memory usage when solving the weighted correlation clustering problem on dense graphs.

Graph		Time (s)		Opt Ratio		Avg. mem. / iter. (GiB)	
	n	Ours	Ruggles et al.	Ours	Ruggles et al.	Ours	Ruggles et al.
CAGrQc	4158	2098	5577	1.33	1.38	4.4	1.3
Power	4941	1393	6082	1.33	1.37	5.9	2
CAHepTh	8638	9660	35021	1.33	1.36	24	8
CAHepPh	11204	71071	135568	1.33	1.46	27.5	15

Experiments: Weighted correlation clustering (dense graphs)



(a) Number of constraints.



(b) Max Violation.

Figure 1: Plots showing the number of constraints returned by the oracle, the number of constraints after the forget step, and the maximum violation of a metric constraint when solving correlation clustering on the Ca-HepTh graph

Experiments: Weighted correlation clustering (sparse graphs)

Table 2: Time taken and quality of solution returned by PROJECT AND FORGET when solving the weighted correlation clustering problem for sparse graphs. The table also displays the number of constraints the traditional LP formulation would have.

Graph	n	# Constraints	Time	Opt Ratio	# Active Constraints	Iters.
Slashdot	82140	5.54×10^{14}	46.7 hours	1.78	384227	145
Epinions	131,828	2.29×10^{15}	121.2 hours	1.77	579926	193

Experiments: Metric nearness

Given D, $n \times n$ matrix of distances, find closest metric

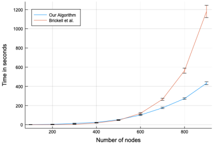
$$\hat{M} = \operatorname{arg\,min} \|D - M\|_p \quad \text{s.t.} \quad M \in \mathrm{MET}_n.$$

Two types of experiments for weighted complete graphs:

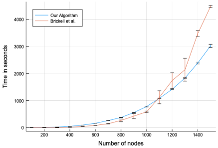
1. Random binary distance matrices
2. Random gaussian distance matrices

Compare against Brickell, et al.

Experiments: Metric nearness



(a) Type one graphs



(b) Type two graphs

Figure 2: Figure showing the average time taken (averaged over 5 trials) by our algorithm and Brickell et al. [6] when solving the metric nearness problem for type 1 and type 2 graphs.

New/different directions: trees and hyperbolic embeddings

Finding a faithful low-dimensional **hyperbolic embedding** key method to extract hierarchical information, learn more representative (?) geometry of data

Examples: analysis of single cell genomic data, linguistics, social network analysis, etc.

Represent data as a tree!

Embed in Euclidean space? NO! Embed in hyperbolic space.

Metric first approach to embeddings

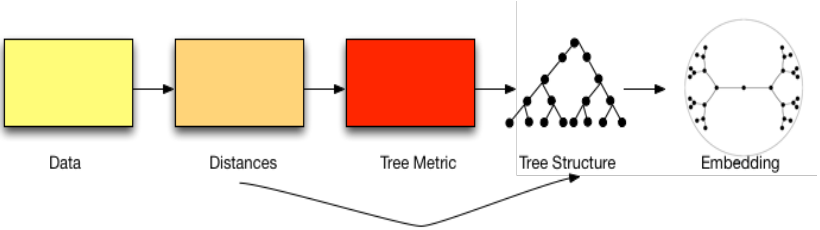
Even simple trees cannot be embedded faithfully in Euclidean space (Linial, et al.)

So, ... recent methods (e.g., Nickel and Kiela, Sala, et al.) learn hyperbolic embeddings instead and then extract hyperbolic metric

Rather than learn a hyperbolic embedding directly, learn a tree structure *first* and then embed tree in $\mathcal{H}^r$.

Metric first: learn an appropriate (tree) metric first and then extract its representation (in hyperbolic space)

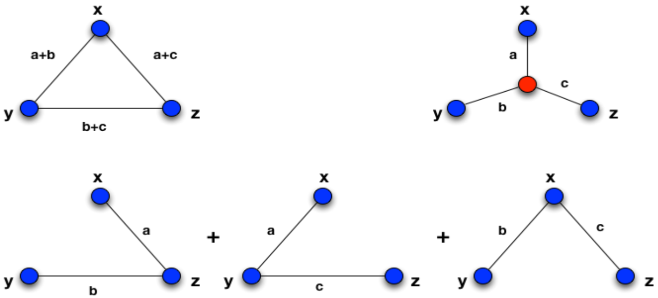
Tree embedding workflow



TreeRep algorithm

Claim

Let N be the number of data points in the data set X and d the tree metric on X. The algorithm TREE STRUCTURE *runs in time $O(N^2)$ in the worst case [conjecture: time $O(N \log N)$ on average, appropriately defined] and produces a tree structure that is consistent with the tree metric d.*



Tree structure, examples