Proximal point algorithm in Hadamard spaces

Miroslav Bacak

Télécom ParisTech

Optimisation Géométrique sur les Variétés
-
Paris, 21 novembre 2014

Contents of the talk

1. Basic facts on Hadamard spaces
2. Proximal point algorithm
3. Applications to computational phylogenetics

Proximal point algorithm in Hadamard spaces

Why? Well… it is used in:

- **Phylogenetics:** computing medians and means of phylogenetic trees.
- **diffusion tensor imaging:** the space $P(n, \mathbb{R})$ of symmetric positive definite matrices $n \times n$ with real entries is a Hadamard space if it is equipped with the Riemannian metric

$$\langle X, Y \rangle_A := \operatorname{Tr}\left(A^{-1} X A^{-1} Y\right), \qquad X, Y \in T_A\left(P(n, \mathbb{R})\right),$$

  for every $A \in P(n, \mathbb{R})$.
- **Computational biology:** shape analyses of tree-like structures:

Tree-like structures in organisms

Figure: Bronchial tubes in lungs

Figure: Transport system in plants

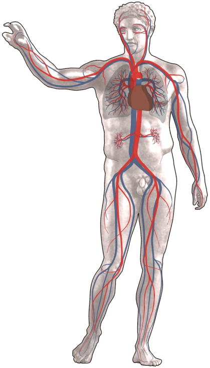


Figure: Human circulatory system

Definition of Hadamard space

Let $(\mathcal{H}, d)$ be a complete metric space where:

1. any two points x_0 and x_1 are connected by a geodesic

$$x\colon [0,1] \to \mathcal{H}\colon t \mapsto x_t,$$

2. and,

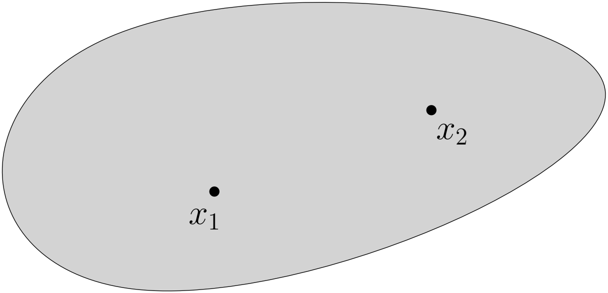
$$d(y, x_t)^2 \leq (1-t)d(y, x_0)^2 + td(y, x_1)^2 - t(1-t)d(x_0, x_1)^2,$$

for every $y \in \mathcal{H}$.

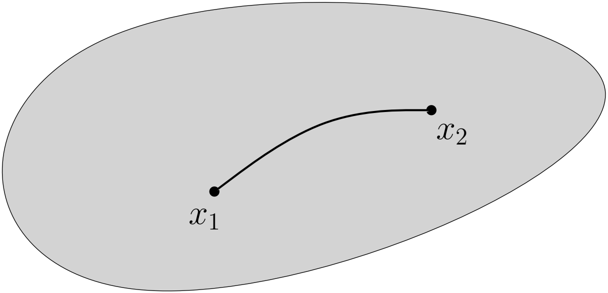
Then $(\mathcal{H}, d)$ is called a Hadamard space.

For today: assume that local compactness.

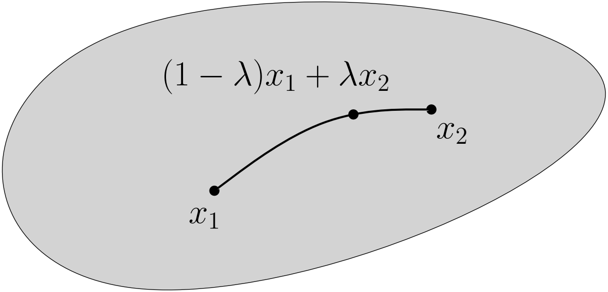
Geodesic space



Geodesic space

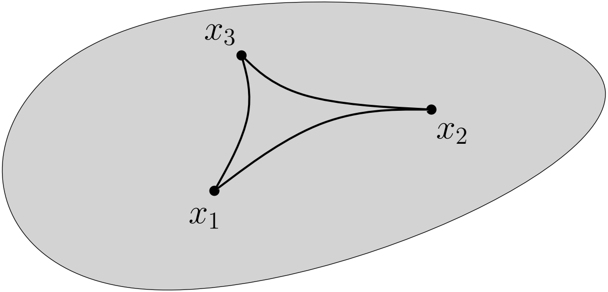


Geodesic space

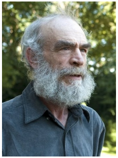


Definition of nonpositive curvature

A geodesic triangle in a geodesic space:



Terminology remark

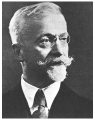


CAT(κ) spaces, for $\kappa \in \mathbb{R}$, were introduced in 1987 by Michail Gromov

C = Cartan

A = Alexandrov

T = Toponogov



We are particularly interested in CAT(0) spaces.

Examples of Hadamard spaces

1. Hilbert spaces, the Hilbert ball
2. complete simply connected Riemannian manifolds with $\mathrm{Sec} \leq 0$
3. $\mathbb{R}$-trees: a metric space T is an $\mathbb{R}$-tree if
   - for $x, y \in T$ there is a unique geodesic $[x, y]$
   - if $[x, y] \cap [y, z] = \{y\}$, then $[x, z] = [x, y] \cup [y, z]$
4. Euclidean buildings
5. the BHV tree space (space of phylogenetic trees)
6. $L^2(M, \mathcal{H})$, where (M, μ) is a probability space:

$$d_2(u, v) := \left(\int_M d\left(u(x), v(x)\right)^2 \, \mathrm{d}\mu(x) \right)^{\frac{1}{2}}, \qquad u, v \in L^2(M, \mathcal{H})$$

Convexity in Hadamard spaces

Let $(\mathcal{H}, d)$ be a Hadamard space. These spaces allow for a natural definition of convexity:

Definition

A set $C \subset \mathcal{H}$ is **convex** if, given $x, y \in C$, we have $[x, y] \subset C$.

Definition

A function $f\colon \mathcal{H} \to (-\infty, \infty]$ is **convex** if $f \circ \gamma$ is a convex function for each geodesic $\gamma\colon [0,1] \to \mathcal{H}$.

Convexity in Hadamard spaces

Let $(\mathcal{H}, d)$ be a Hadamard space. These spaces allow for a natural definition of convexity:

Definition

A set $C \subset \mathcal{H}$ is **convex** if, given $x, y \in C$, we have $[x, y] \subset C$.

Definition

A function $f\colon \mathcal{H} \to (-\infty, \infty]$ is **convex** if $f \circ \gamma$ is a convex function for each geodesic $\gamma\colon [0,1] \to \mathcal{H}$.

Convexity in Hadamard spaces

Let $(\mathcal{H}, d)$ be a Hadamard space. These spaces allow for a natural definition of convexity:

Definition

A set $C \subset \mathcal{H}$ is **convex** if, given $x, y \in C$, we have $[x, y] \subset C$.

Definition

A function $f\colon \mathcal{H} \to (-\infty, \infty]$ is **convex** if $f \circ \gamma$ is a convex function for each geodesic $\gamma\colon [0,1] \to \mathcal{H}$.

Examples of convex functions

1. The **indicator function** of a convex closed set $C \subset \mathcal{H}$:

$$\iota_C(x) := 0, \text{ if } x \in C, \quad \text{and} \quad \iota_C(x) := \infty, \text{ if } x \notin C.$$

2. The **distance function** to a closed convex subset $C \subset \mathcal{H}$:

$$d_C(x) := \inf_{c \in C} d(x, c), \quad x \in \mathcal{H}.$$

3. The **displacement function** of an isometry $T \colon \mathcal{H} \to \mathcal{H}$:

$$\delta_T(x) := d(x, Tx), \quad x \in \mathcal{H}.$$

Examples of convex functions

1. The **indicator function** of a convex closed set $C \subset \mathcal{H}$:

$$\iota_C(x) := 0, \text{ if } x \in C, \quad \text{and} \quad \iota_C(x) := \infty, \text{ if } x \notin C.$$

2. The **distance function** to a closed convex subset $C \subset \mathcal{H}$:

$$d_C(x) := \inf_{c \in C} d(x, c), \quad x \in \mathcal{H}.$$

3. The **displacement function** of an isometry $T \colon \mathcal{H} \to \mathcal{H}$:

$$\delta_T(x) := d(x, Tx), \quad x \in \mathcal{H}.$$

Examples of convex functions

1. The **indicator function** of a convex closed set $C \subset \mathcal{H}$:

$$\iota_C(x) := 0, \text{ if } x \in C, \quad \text{and} \quad \iota_C(x) := \infty, \text{ if } x \notin C.$$

2. The **distance function** to a closed convex subset $C \subset \mathcal{H}$:

$$d_C(x) := \inf_{c \in C} d(x, c), \quad x \in \mathcal{H}.$$

3. The **displacement function** of an isometry $T \colon \mathcal{H} \to \mathcal{H}$:

$$\delta_T(x) := d(x, Tx), \quad x \in \mathcal{H}.$$

Examples of convex functions

4. Let $c\colon [0,\infty) \to \mathcal{H}$ be a geodesic ray. The function $b_c\colon \mathcal{H} \to \mathbb{R}$ defined by

$$b_c(x) := \lim_{t\to\infty} \left[d\left(x, c(t)\right) - t\right], \quad x \in \mathcal{H},$$

is called the **Busemann function** associated to the ray c.

5. The **energy** of a mapping $u\colon M \to \mathcal{H}$ given by

$$E(u) := \iint_{M\times M} d\left(u(x), u(y)\right)^2 p(x, \mathrm{d}y)\mathrm{d}\mu(x),$$

where (M, μ) is a measure space with a Markov kernel $p(x, \mathrm{d}y)$.

E is convex continuous on $L^2(M, \mathcal{H})$.

Examples of convex functions

4. Let $c\colon [0,\infty) \to \mathcal{H}$ be a geodesic ray. The function $b_c\colon \mathcal{H} \to \mathbb{R}$ defined by

$$b_c(x) := \lim_{t\to\infty} \left[d\left(x, c(t)\right) - t \right], \quad x \in \mathcal{H},$$

   is called the **Busemann function** associated to the ray c.

5. The **energy** of a mapping $u\colon M \to \mathcal{H}$ given by

$$E(u) := \iint_{M\times M} d\left(u(x), u(y)\right)^2 p(x, \mathrm{d}y)\mathrm{d}\mu(x),$$

   where (M,μ) is a measure space with a Markov kernel $p(x, \mathrm{d}y)$.

   E is convex continuous on $L^2(M,\mathcal{H})$.

Examples of convex functions

6. Given $a_1, \ldots, a_N \in \mathcal{H}$ and $w_1, \ldots, w_N > 0$, set

$$f(x) := \sum_{n=1}^{N} w_n d\left(x, a_n\right)^p, \qquad x \in \mathcal{H},$$

where $p \in [1, \infty)$.

- If $p = 1$, we get **Fermat-Weber problem** for optimal facility location. A minimizer of f is called a **median.**
- If $p = 2$, then a minimizer of f is the **barycenter** of

$$\mu := \sum_{n=1}^{N} w_n \delta_{a_n},$$

or the **mean** of $a_1, \ldots, a_N$.

Examples of convex functions

6. Given $a_1, \ldots, a_N \in \mathcal{H}$ and $w_1, \ldots, w_N > 0$, set

$$f(x) := \sum_{n=1}^{N} w_n d(x, a_n)^p, \qquad x \in \mathcal{H},$$

where $p \in [1, \infty)$.

- If $p = 1$, we get **Fermat-Weber problem** for optimal facility location. A minimizer of f is called a **median.**
- If $p = 2$, then a minimizer of f is the **barycenter** of

$$\mu := \sum_{n=1}^{N} w_n \delta_{a_n},$$

or the **mean** of $a_1, \ldots, a_N$.

Strongly convex functions

A function $f\colon \mathcal{H} \to (-\infty, \infty]$ is **strongly convex** with parameter $\beta > 0$ if

$$f\left((1-t)x + ty\right) \leq (1-t)f(x) + tf(y) - \beta t(1-t)d(x,y)^2,$$

for any $x, y \in \mathcal{H}$ and $t \in [0,1]$.

Each strongly has a unique minimizer.

Example

Given $y \in \mathcal{H}$, the function $f := d(y,\cdot)^2$ is strongly convex. Indeed,

$$d\left(y, x_t\right)^2 \leq (1-t)d\left(y, x_0\right)^2 + td\left(y, x_1\right)^2 - t(1-t)d\left(x_0, x_1\right)^2,$$

for each geodesic $x\colon [0,1] \to \mathcal{H}$.

Proximal point algorithm

Let $f\colon \mathcal{H} \to (-\infty, \infty]$ be convex lsc.

Optimization problem: $\min_{x \in \mathcal{H}} f(x).$

Recall: no (sub)differential, no shooting (singularities).

Implicit methods are appropriate. The PPA generates a sequence

$$x_i := J_{\lambda_i}(x_{i-1}) := \arg\min_{y \in \mathcal{H}} \left[f(y) + \frac{1}{2\lambda_i} d(y, x_{i-1})^2 \right],$$

where $x_0 \in \mathcal{H}$ is a given starting point and $\lambda_i > 0$, for each $i \in \mathbb{N}$.

Proximal point algorithm

Let $f\colon \mathcal{H} \to (-\infty, \infty]$ be convex lsc.

Optimization problem: $\min_{x \in \mathcal{H}} f(x).$

Recall: no (sub)differential, no shooting (singularities).

Implicit methods are appropriate. The PPA generates a sequence

$$x_i := J_{\lambda_i}(x_{i-1}) := \arg\min_{y \in \mathcal{H}} \left[f(y) + \frac{1}{2\lambda_i} d(y, x_{i-1})^2 \right],$$

where $x_0 \in \mathcal{H}$ is a given starting point and $\lambda_i > 0$, for each $i \in \mathbb{N}$.

Convergence of proximal point algorithm

Theorem (M.B., 2011)

Let $f\colon \mathcal{H} \to (-\infty, \infty]$ be a convex lsc function attaining its minimum. Given $x_0 \in \mathcal{H}$ and (λ_i) such that $\sum_1^\infty \lambda_i = \infty$, the PPA sequence (x_i) converges to a minimizer of f.

(Resolvents are firmly nonexpansive - cheap version for $\lambda_i = \lambda$.)

Disadvantage: The resolvents

$$x_i := J_{\lambda_i}(x_{i-1}) := \underset{y \in \mathcal{H}}{\arg\min} \left[f(y) + \frac{1}{2\lambda_i} d(y, x_{i-1})^2 \right],$$

are often difficult to compute.

Convergence of proximal point algorithm

Theorem (M.B., 2011)

Let $f\colon \mathcal{H} \to (-\infty, \infty]$ be a convex lsc function attaining its minimum. Given $x_0 \in \mathcal{H}$ and (λ_i) such that $\sum_1^\infty \lambda_i = \infty$, the PPA sequence (x_i) converges to a minimizer of f.

(Resolvents are firmly nonexpansive - cheap version for $\lambda_i = \lambda$.)

Disadvantage: The resolvents

$$x_i := J_{\lambda_i}(x_{i-1}) := \operatorname*{arg\,min}_{y \in \mathcal{H}} \left[f(y) + \frac{1}{2\lambda_i} d(y, x_{i-1})^2 \right],$$

are often difficult to compute.

Splitting proximal point algorithm

Let $f_1, \ldots, f_N$ be convex lsc and consider

$$f(x) := \sum_{n=1}^{N} f_n(x), \qquad x \in \mathcal{H}.$$

Example (Median and mean)

$$f_n := d(\cdot, a_n), \qquad f_n := d(\cdot, a_n)^2.$$

Key idea: apply resolvents J_λ^n's of f_n's in a cyclic or random order.

Splitting proximal point algorithm

Let $f_1, \dots, f_N$ be convex lsc and consider

$$f(x) := \sum_{n=1}^{N} f_n(x), \qquad x \in \mathcal{H}.$$

Example (Median and mean)

$$f_n := d(\cdot, a_n), \qquad f_n := d(\cdot, a_n)^2.$$

Key idea: apply resolvents J_λ^n's of f_n's in a cyclic or random order.

Splitting proximal point algorithm

Let $f_1, \dots, f_N$ be convex lsc and consider

$$f(x) := \sum_{n=1}^{N} f_n(x), \qquad x \in \mathcal{H}.$$

Example (Median and mean)

$$f_n := d(\cdot, a_n), \qquad f_n := d(\cdot, a_n)^2.$$

Key idea: apply resolvents J_λ^n's of f_n's in a cyclic or random order.

Splitting proximal point algorithm

Let $x_0 \in \mathcal{H}$ be a starting point. For each $k \in \mathbb{N}_0$ we apply resolvents in **cyclic order:**

$$
\begin{aligned}
x_{kN+1} &:= J^1_{\lambda_k}\left(x_{kN}\right), \\
x_{kN+2} &:= J^2_{\lambda_k}\left(x_{kN+1}\right), \\
&\vdots \\
x_{kN+N} &:= J^N_{\lambda_k}\left(x_{kN+N-1}\right),
\end{aligned}
$$

or in **random order:**

$$
x_{i+1} := J^{r_i}_{\lambda_i}\left(x_i\right),
$$

where (r_i) are random variables with values in $\{1, \ldots, N\}$.

Convergence of splitting proximal point algorithm

Theorem (Cyclic order version + Random order version)

Assume that f_n are Lipschitz (or locally Lipschitz and the minimizing sequence is bounded). Then

1. *the cyclic PPA sequence converges to a minimizer of f*
2. *the random PPA sequence converges to a minimizer of f almost surely.*

Assumptions are satisfied for

$$f(x) := \sum_{n=1}^{N} w_n d(x, a_n)^p, \qquad x \in \mathcal{H},$$

where $p \in [1, \infty)$.

Convergence of splitting proximal point algorithm

Theorem (Cyclic order version + Random order version)

Assume that f_n are Lipschitz (or locally Lipschitz and the minimizing sequence is bounded). Then

1. *the cyclic PPA sequence converges to a minimizer of f*
2. *the random PPA sequence converges to a minimizer of f almost surely.*

Assumptions are satisfied for

$$f(x) := \sum_{n=1}^{N} w_n d(x, a_n)^p, \qquad x \in \mathcal{H},$$

where $p \in [1, \infty)$.

Splitting proximal point algorithm (for mean)

Hence instead of computing (the usual PPA)

$$x_{i+1} := \arg\min_{z \in \mathcal{H}} \left[\sum_{n=1}^{N} d(z, a_n)^2 + \frac{1}{2\lambda_i} d(z, x_i)^2 \right],$$

we are to minimize the function

$$x_{i+1} := \arg\min_{z \in \mathcal{H}} \left[d(z, a_n)^2 + \frac{1}{2\lambda_i} d(z, x_i)^2 \right],$$

where a_n are chosen in a cyclic or random order.

This is **one-dimensional** problem!

$\Longrightarrow \quad x_{i+1}$ is a convex combination of a_n and x_i.

Splitting proximal point algorithm (for mean)

Hence instead of computing (the usual PPA)

$$x_{i+1} := \arg\min_{z\in\mathcal{H}} \left[\sum_{n=1}^{N} d\left(z, a_n\right)^2 + \frac{1}{2\lambda_i} d\left(z, x_i\right)^2 \right],$$

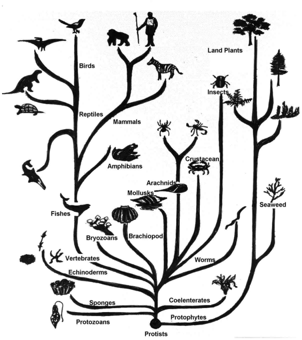
we are to minimize the function

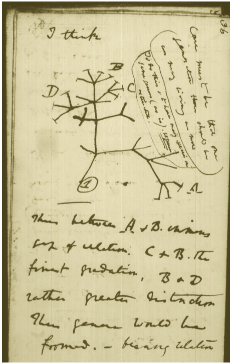
$$x_{i+1} := \arg\min_{z\in\mathcal{H}} \left[d\left(z, a_n\right)^2 + \frac{1}{2\lambda_i} d\left(z, x_i\right)^2 \right],$$

where a_n are chosen in a cyclic or random order.

This is **one-dimensional** problem!

$\Longrightarrow \quad x_{i+1}$ is a convex combination of a_n and x_i.





Left: One of many evolutionary trees

Right: A picture of an evolutionary tree by Charles Darwin (1837)

Billera-Holmes-Vogtmann tree space $\mathcal{T}_d$

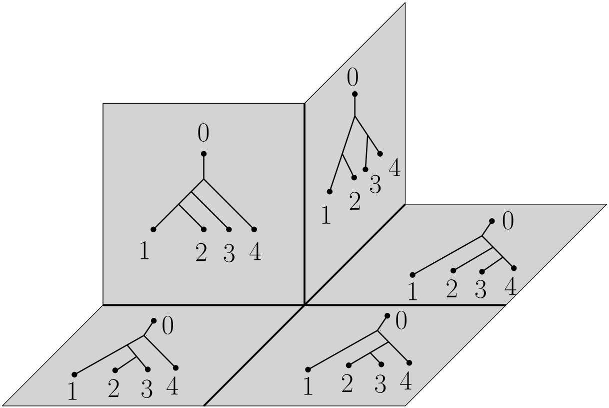


Figure: 5 out of 15 orthants of $\mathcal{T}_4$

Billera-Holmes-Vogtmann tree space $\mathcal{T}_d$

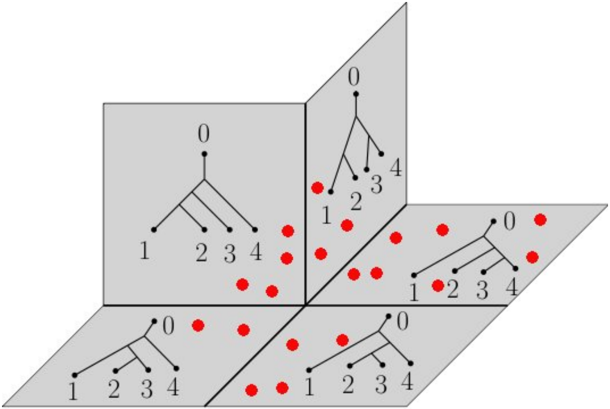


Figure: A finite set of trees in $\mathcal{T}_4$

Billera-Holmes-Vogtmann tree space $\mathcal{T}_d$

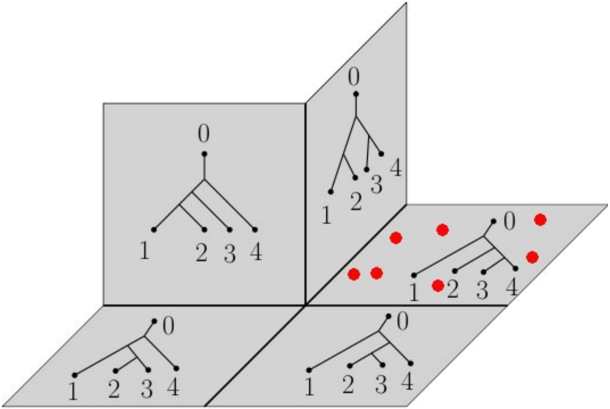


Figure: Consider the most frequent tree topology only

Computing the mean: Random order version

Algorithm (SPPA with $f_n := d(\cdot, T_n)^2$)

Input: $T_1, \ldots, T_N \in \mathcal{T}_d$

Step 1: $S_1 := T_1$ *and* $i := 1$

Step 2: *choose* $r \in \{1, \ldots, N\}$ *at random*

Step 3: $S_{i+1} := \frac{1}{i+1} T_r + \frac{i}{i+1} S_i$

Step 4: $i := i + 1$

Step 5: *go to Step 2*

Geodesics can be computed in **polynomial** time:

The Owen-Provan algorithm (2011)

Computing the mean: Random order version

Algorithm (SPPA with $f_n := d(\cdot, T_n)^2$)

Input: $T_1, \dots, T_N \in \mathcal{T}_d$

Step 1: $S_1 := T_1$ *and* $i := 1$

Step 2: *choose* $r \in \{1, \dots, N\}$ *at random*

Step 3: $S_{i+1} := \frac{1}{i+1} T_r + \frac{i}{i+1} S_i$

Step 4: $i := i + 1$

Step 5: *go to Step 2*

Geodesics can be computed in **polynomial** time:

The Owen-Provan algorithm (2011)

Computing the mean: Random order version

Algorithm (SPPA with $f_n := d(\cdot, T_n)^2$)

Input: $T_1, \ldots, T_N \in \mathcal{T}_d$

Step 1: $S_1 := T_1$ *and* $i := 1$

Step 2: *choose* $r \in \{1, \ldots, N\}$ *at random*

Step 3: $S_{i+1} := \frac{1}{i+1} T_r + \frac{i}{i+1} S_i$

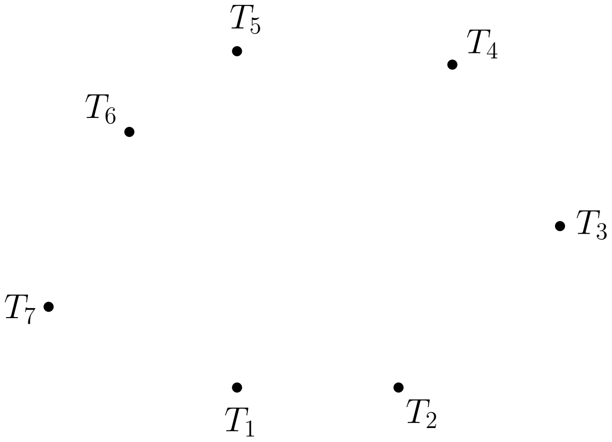
Step 4: $i := i + 1$

Step 5: *go to Step 2*

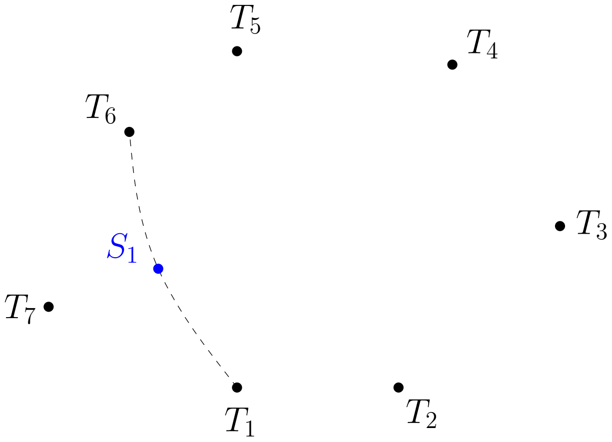
Geodesics can be computed in **polynomial** time:

The Owen-Provan algorithm (2011)

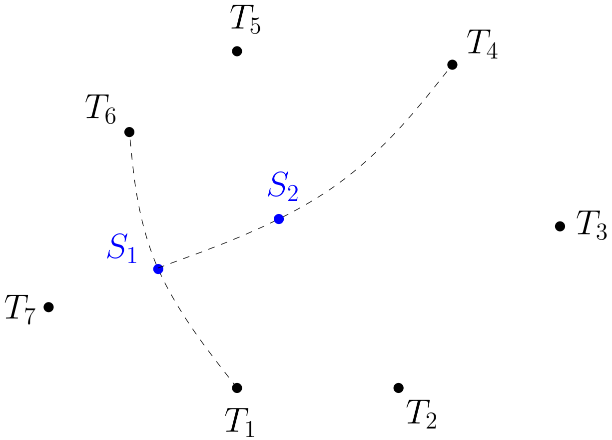
Computing the mean: Random order version



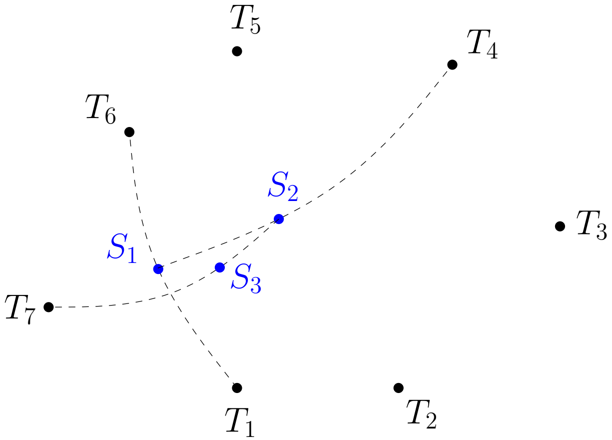
Computing the mean: Random order version



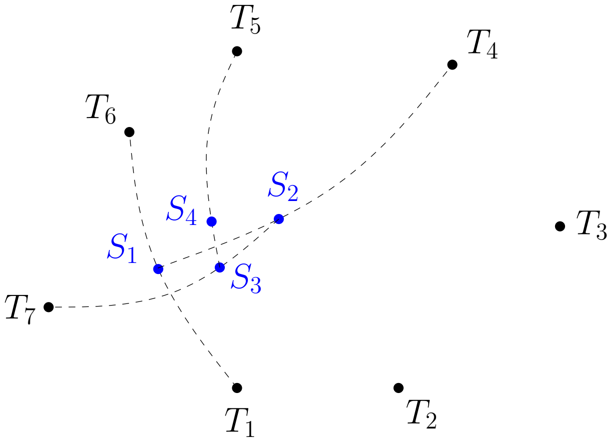
Computing the mean: Random order version



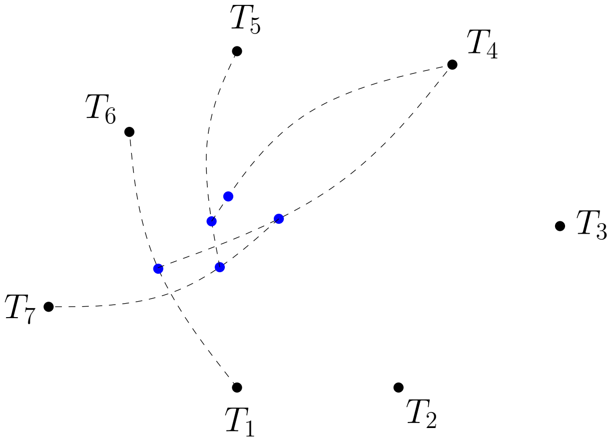
Computing the mean: Random order version



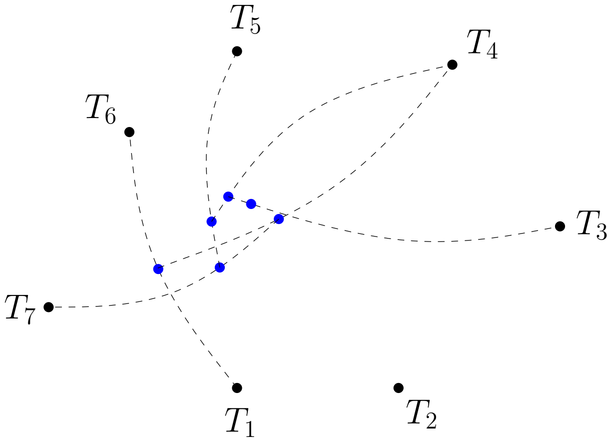
Computing the mean: Random order version



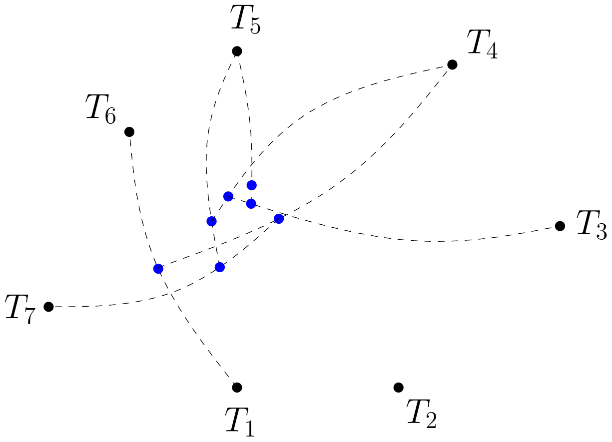
Computing the mean: Random order version



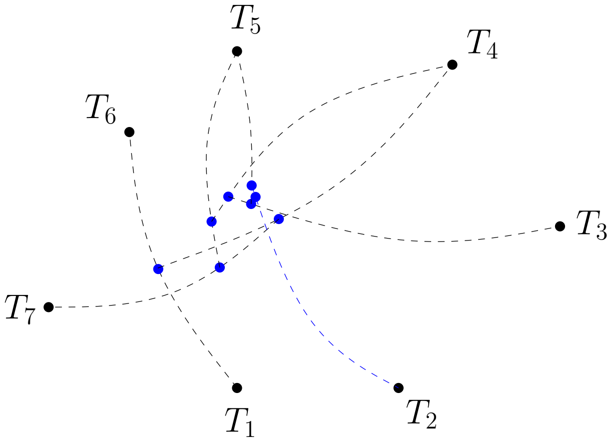
Computing the mean: Random order version



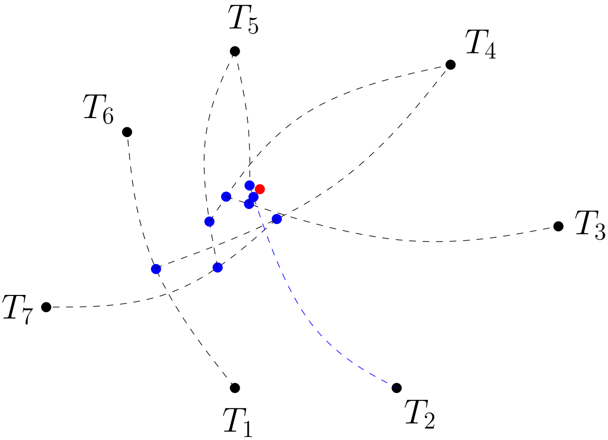
Computing the mean: Random order version



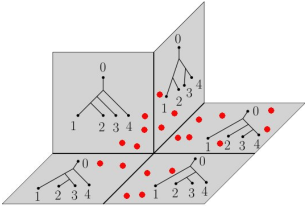
Computing the mean: Random order version



Computing the mean: Random order version



Le Big Data



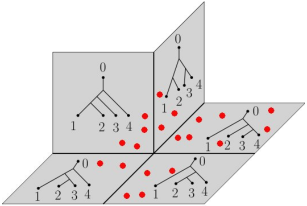
Space $\mathcal{T}_d$: orthant dimension $= d-2$, # of orthants $= (2d-3)!!$

The actual dimension of $\mathcal{T}_d$ is $d+1+(d-2)(2d-3)!!$

# of trees $= N$ (e.g. coming from an MCMC simulation)

Our computation: $d = 12$ hence $\dim \approx 10^{11}$ and $N = 10^5$

Le Big Data



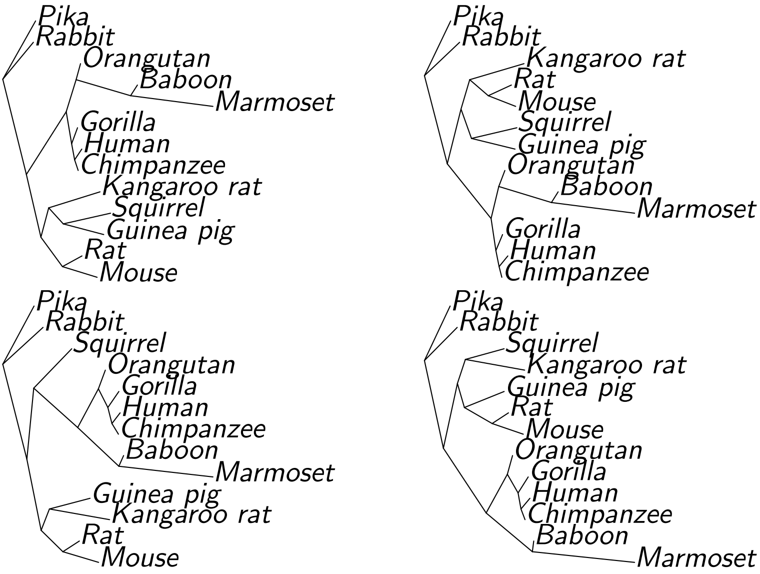
Space $\mathcal{T}_d$: orthant dimension $= d-2$, $\#$ of orthants $= (2d-3)!!$

The actual dimension of $\mathcal{T}_d$ is $d+1+(d-2)(2d-3)!!$

$\#$ of trees $= N$ (e.g. coming from an MCMC simulation)

Our computation: $d = 12$ hence $\dim \approx 10^{11}$ and $N = 10^5$

Resuls (courtesy of Philipp Benner)



Resuls (courtesy of Philipp Benner) . . . continued.

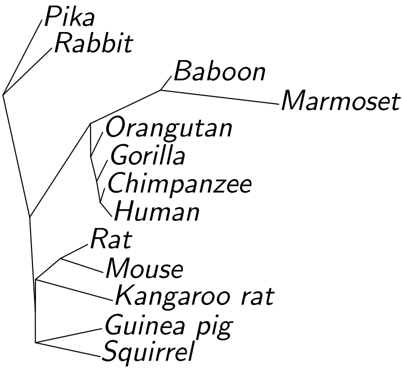


Figure: Approximation of the mean of the 100,000 trees.

References

1. **M.B.:** *Computing medians and means in Hadamard spaces,* SIAM J. Optim. 24 (2014), no. 3, 1542–1566.
2. **Benner, Bacak, Bourguignon:** *Point estimates in phylogenetic reconstructions,* Bioinformatics, Vol 30 (2014), Issue 17.
3. **M.B.:** *Convex analysis and optimization in Hadamard spaces,* De Gruyter, Berlin, 2014.