The Minimum Barrier Distance Transform

Krzysztof Chris Ciesielski

Department of Mathematics, West Virginia University
and
MIPG, Department of Radiology, University of Pennsylvania

MIPG Seminar, UPenn, November 16, 2012

Outline

1. Minimum Barrier Distance, $\hat{\rho}$, in the discrete setting
2. How to compute $\hat{\rho}$?
3. Minimum Barrier Distance, ρ, in the continuous setting
4. Experiments: comparison with other distance measures
5. Newest result: fast algorithm for computing $\hat{\rho}$

Outline

1. Minimum Barrier Distance, $\hat{\rho}$, in the discrete setting
2. How to compute $\hat{\rho}$?
3. Minimum Barrier Distance, ρ, in the continuous setting
4. Experiments: comparison with other distance measures
5. Newest result: fast algorithm for computing $\hat{\rho}$

Image, scene, and the associated graph

$\widehat{f}\colon \widehat{D} \to \mathbb{R}$ is a digital image, where

$\widehat{D} = \mathbb{Z}^k \cap \prod_{i=1}^{k} [a_i, b_i]$ ($a_i, b_i \in \mathbb{R}$) is a digital scene

with $x, y \in \widehat{D}$ adjacent provided $\sum_i |x(i) - y(i)| = 1$.

We will treat also this structure,

$G = \langle \widehat{D}, \{\{x, y\} \colon x, y \text{ adjacent}\}, \widehat{f} \rangle$,

as a vertex weighted graph $G = \langle V(G), E(G), \widehat{w} \rangle$.

Image, scene, and the associated graph

$\widehat{f}\colon \widehat{D} \to \mathbb{R}$ is a digital image, where

$\widehat{D} = \mathbb{Z}^k \cap \prod_{i=1}^{k} [a_i, b_i]$ ($a_i, b_i \in \mathbb{R}$) is a digital scene

with $x, y \in \widehat{D}$ adjacent provided $\sum_i |x(i) - y(i)| = 1$.

We will treat also this structure,

$G = \langle \widehat{D}, \{\{x, y\}\colon x, y \text{ adjacent}\}, \widehat{f} \rangle$,

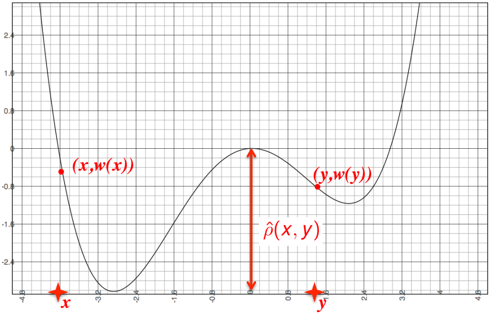
as a vertex weighted graph $G = \langle V(G), E(G), \widehat{w} \rangle$.

Minimum Barrier Distance in discrete setting

For a path $p = \langle c_1, \ldots, c_k \rangle$ in $G = \langle \widehat{D}, E, \widehat{w} \rangle$

$$c_b(p) = \max_i \widehat{w}(c_i) - \min_i \widehat{w}(c_i)$$

is the barrier cost of p.



The barrier distance between x and y in $\widehat{D}$ is given by:

$$\hat{\rho}(x, y) = \min\{c_b(p) \colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$$

Minimum Barrier Distance in discrete setting

For a path $p = \langle c_1, \ldots, c_k \rangle$ in $G = \langle \widehat{D}, E, \widehat{w} \rangle$

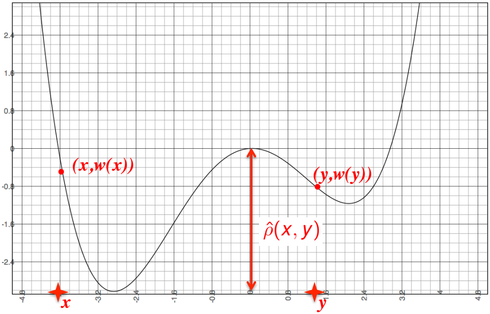
$$c_b(p) = \max_i \widehat{w}(c_i) \; - \min_i \widehat{w}(c_i)$$

is the barrier cost of p.

The barrier distance

between x and y in $\widehat{D}$

is given by:



$$\hat{\rho}(x, y) = \min\{c_b(p) \colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$$

MBD vs geodesic distance

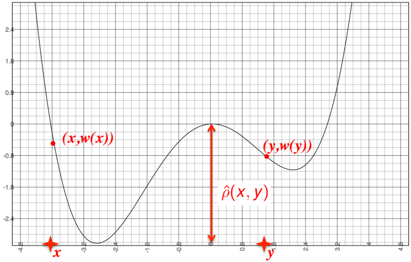
$$\hat{\rho}(x,y) = \min\{c_b(p)\colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$$

$\hat{\rho}(x,y)$ is, in a way,

a vertical component of

the geodesic distance

between x and y.



Theorem

$\hat{\rho}$ is a pseudo-metric:
it is symmetric and it satisfies the triangle inequality.
(However, $\hat{\rho}(x,y)$ can be equal 0 for $x \neq y$.)

MBD vs geodesic distance

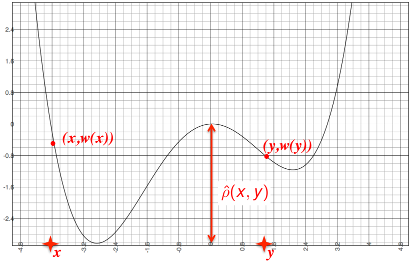
$$\hat{\rho}(x, y) = \min\{c_b(p) \colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$$

$\hat{\rho}(x, y)$ is, in a way,

a vertical component of

the geodesic distance

between x and y.



Theorem

$\hat{\rho}$ is a pseudo-metric:
it is symmetric and it satisfies the triangle inequality.
(However, $\hat{\rho}(x, y)$ can be equal 0 for $x \neq y$.)

MBD as a measure of connectivity

$\hat{\rho}(x, y) = \min\{c_b(p) \colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$

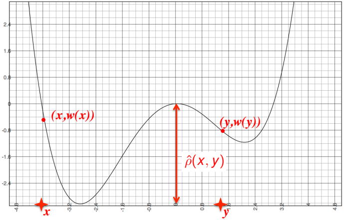
$\beta(x, y) = \exp(-\hat{\rho}(x, y))$

has some similarity to the

FC connectivity measure for the

object-feature base affinity with

average intensity value $m = \widehat{w}(s)$:



$\beta(x, s)$ is small when $|\widehat{w}(x) - m|$ is large.

$\hat{\rho}(x, y)$ can be used to define RFC-like object:

$$P(s, t) = \{c \in \widehat{D} \colon \hat{\rho}(c, s) < \hat{\rho}(c, t)\}.$$

(Not studied yet.)

MBD as a measure of connectivity

$\hat{\rho}(x, y) = \min\{c_b(p) \colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$

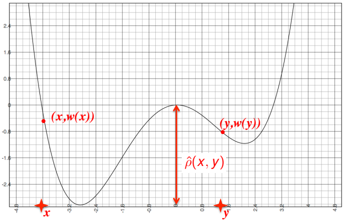
$\beta(x, y) = \exp(-\hat{\rho}(x, y))$

has some similarity to the

FC connectivity measure for the

object-feature base affinity with

average intensity value $m = \widehat{w}(s)$:



$\beta(x, s)$ is small when $|\widehat{w}(x) - m|$ is large.

$\hat{\rho}(x, y)$ can be used to define RFC-like object:

$$P(s, t) = \{c \in \widehat{D} \colon \hat{\rho}(c, s) < \hat{\rho}(c, t)\}.$$

(Not studied yet.)

MBD as a measure of connectivity

$\hat{\rho}(x, y) = \min\{c_b(p) \colon p$ is a path in G from x to $y\}$

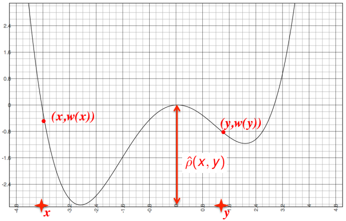
$\beta(x, y) = \exp(-\hat{\rho}(x, y))$

has some similarity to the

FC connectivity measure for the

object-feature base affinity with

average intensity value $m = \widehat{w}(s)$:



$\beta(x, s)$ is small when $|\widehat{w}(x) - m|$ is large.

$\hat{\rho}(x, y)$ can be used to define RFC-like object:

$$P(s, t) = \{c \in \widehat{D} \colon \hat{\rho}(c, s) < \hat{\rho}(c, t)\}.$$

(Not studied yet.)

MBD as a measure of connectivity

$\hat{\rho}(x, y) = \min\{c_b(p) \colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$

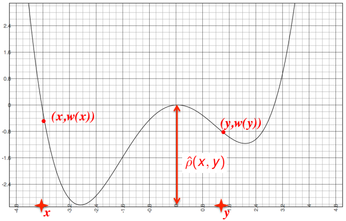
$\beta(x, y) = \exp(-\hat{\rho}(x, y))$

has some similarity to the

FC connectivity measure for the

object-feature base affinity with

average intensity value $m = \widehat{w}(s)$:



$\beta(x, s)$ is small when $|\widehat{w}(x) - m|$ is large.

$\hat{\rho}(x, y)$ can be used to define RFC-like object:

$$P(s, t) = \{c \in \widehat{D} \colon \hat{\rho}(c, s) < \hat{\rho}(c, t)\}.$$

(Not studied yet.)

MBD as a measure of connectivity

$\hat{\rho}(x, y) = \min\{c_b(p) \colon p$ is a path in G from x to $y\}$

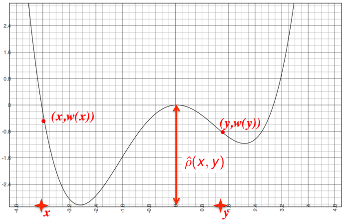
$\beta(x, y) = \exp(-\hat{\rho}(x, y))$

has some similarity to the

FC connectivity measure for the

object-feature base affinity with

average intensity value $m = \widehat{w}(s)$:



$\beta(x, s)$ is small when $|\widehat{w}(x) - m|$ is large.

$\hat{\rho}(x, y)$ can be used to define RFC-like object:

$$P(s, t) = \{c \in \widehat{D} \colon \hat{\rho}(c, s) < \hat{\rho}(c, t)\}.$$

(Not studied yet.)

Outline

1. Minimum Barrier Distance, $\hat{\rho}$, in the discrete setting
2. How to compute $\hat{\rho}$?
3. Minimum Barrier Distance, ρ, in the continuous setting
4. Experiments: comparison with other distance measures
5. Newest result: fast algorithm for computing $\hat{\rho}$

Can Dijkstra-like algorithm find $\hat{\rho}(x, y)$?

Natural Algorithm:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: $L(y), U(y) \in \mathbb{R}$, a path p_y from x to y with the range in $[L(y), U(y)]$ s.t. (hopefully) $\hat{\rho}(x, y) = U(y) - L(y)$.

Initialization: Push x to queue Q ordered via $U(y) - L(y)$.

Put $L(y) = -\infty$, $U(y) = \infty$ for $y \neq x$, $L(x) = U(x) = \widehat{w}(x)$;
while Q is not empty **do**
    Pop z from Q;
    for every y adjacent to z **do**
        Put $L = \min\{L(z), \widehat{w}(y)\}$ and $U = \max\{U(z), \widehat{w}(y)\}$;
        if $U(y) - L(y) > U - L$ **then**
            Put $L(y) = L$ and $U(y) = U$;
            Push y to Q;
        end if
    end for
end while

Can Dijkstra-like algorithm find $\hat{\rho}(x, y)$?

Natural Algorithm:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: $L(y), U(y) \in \mathbb{R}$, a path p_y from x to y with the range in $[L(y), U(y)]$ s.t. (hopefully) $\hat{\rho}(x, y) = U(y) - L(y)$.

Initialization: Push x to queue Q ordered via $U(y) - L(y)$.

Put $L(y) = -\infty$, $U(y) = \infty$ for $y \neq x$, $L(x) = U(x) = \widehat{w}(x)$;
while Q is not empty **do**
  Pop z from Q;
  for every y adjacent to z **do**
    Put $L = \min\{L(z), \widehat{w}(y)\}$ and $U = \max\{U(z), \widehat{w}(y)\}$;
    if $U(y) - L(y) > U - L$ **then**
      Put $L(y) = L$ and $U(y) = U$;
      Push y to Q;
    end if
  end for
end while

Can Dijkstra-like algorithm find $\hat{\rho}(x, y)$?

Natural Algorithm:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: $L(y), U(y) \in \mathbb{R}$, a path p_y from x to y with the range in $[L(y), U(y)]$ s.t. (hopefully) $\hat{\rho}(x, y) = U(y) - L(y)$.

Initialization: Push x to queue Q ordered via $U(y) - L(y)$.

```
Put L(y) = −∞, U(y) = ∞ for y ≠ x, L(x) = U(x) = ŵ(x);
while Q is not empty do
    Pop z from Q;
    for every y adjacent to z do
        Put L = min{L(z), ŵ(y)} and U = max{U(z), ŵ(y)};
        if U(y) − L(y) > U − L then
            Put L(y) = L and U(y) = U;
            Push y to Q;
        end if
    end for
end while
```

Can Dijkstra-like algorithm find $\hat{\rho}(x, y)$?

Natural Algorithm:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: $L(y), U(y) \in \mathbb{R}$, a path p_y from x to y with the range in $[L(y), U(y)]$ s.t. (hopefully) $\hat{\rho}(x, y) = U(y) - L(y)$.

Initialization: Push x to queue Q ordered via $U(y) - L(y)$.

```
Put L(y) = −∞, U(y) = ∞ for y ≠ x, L(x) = U(x) = ŵ(x);
while Q is not empty do
    Pop z from Q;
    for every y adjacent to z do
        Put L = min{L(z), ŵ(y)} and U = max{U(z), ŵ(y)};
        if U(y) − L(y) > U − L then
            Put L(y) = L and U(y) = U;
            Push y to Q;
        end if
    end for
end while
```

Can Dijkstra-like algorithm find $\hat{\rho}(x, y)$?

Natural Algorithm:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: $L(y), U(y) \in \mathbb{R}$, a path p_y from x to y with the range in $[L(y), U(y)]$ s.t. (hopefully) $\hat{\rho}(x, y) = U(y) - L(y)$.

Initialization: Push x to queue Q ordered via $U(y) - L(y)$.

```
Put L(y) = −∞, U(y) = ∞ for y ≠ x, L(x) = U(x) = ŵ(x);
while Q is not empty do
    Pop z from Q;
    for every y adjacent to z do
        Put L = min{L(z), ŵ(y)} and U = max{U(z), ŵ(y)};
        if U(y) − L(y) > U − L then
            Put L(y) = L and U(y) = U;
            Push y to Q;
        end if
    end for
end while
```

Can Dijkstra-like algorithm find $\hat{\rho}(x, y)$?

Natural Algorithm:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: $L(y), U(y) \in \mathbb{R}$, a path p_y from x to y with the range in $[L(y), U(y)]$ s.t. (hopefully) $\hat{\rho}(x, y) = U(y) - L(y)$.

Initialization: Push x to queue Q ordered via $U(y) - L(y)$.

```
Put L(y) = −∞, U(y) = ∞ for y ≠ x, L(x) = U(x) = ŵ(x);
while Q is not empty do
    Pop z from Q;
    for every y adjacent to z do
        Put L = min{L(z), ŵ(y)} and U = max{U(z), ŵ(y)};
        if U(y) − L(y) > U − L then
            Put L(y) = L and U(y) = U;
            Push y to Q;
        end if
    end for
end while
```

Failure of Natural Algorithm!

1	0.8
0.41	0.5
0.5	0.6

Let x be the lower left spel, with $\widehat{w} = 0.5$.

Let y be the upper right spel, with $\widehat{w} = 0.8$.

The optimal path is $p_o = \langle .5; .6; .5; .8 \rangle$,

giving $\hat{\rho}(x, y) = .3$

However, the Natural Algorithm returns

the path $p = \langle .5; .41; .5; .8 \rangle$

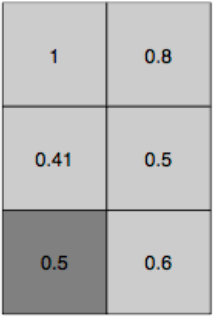
with $c_b(p) = .39 > \hat{\rho}(x, y)$.

Question (We do not know the answer)

Do the numbers $U(y) - L(y)$ returned by Natural Algorithm approximate $\hat{\rho}(x, y)$ in any reasonable sense?

So, how do we effectively compute the numbers $\hat{\rho}(x, y)$?

Failure of Natural Algorithm!



Let x be the lower left spel, with $\widehat{w} = 0.5$.

Let y be the upper right spel, with $\widehat{w} = 0.8$.

The optimal path is $p_o = \langle .5; .6; .5; .8 \rangle$,

giving $\hat{\rho}(x, y) = .3$

However, the Natural Algorithm returns

the path $p = \langle .5; .41; .5; .8 \rangle$

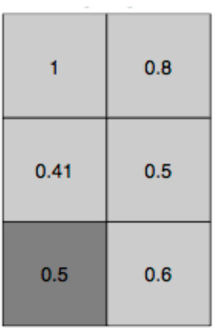
with $c_b(p) = .39 > \hat{\rho}(x, y)$.

Question (We do not know the answer)

Do the numbers $U(y) - L(y)$ returned by Natural Algorithm approximate $\hat{\rho}(x, y)$ in any reasonable sense?

So, how do we effectively compute the numbers $\hat{\rho}(x, y)$?

Failure of Natural Algorithm!



Let x be the lower left spel, with $\widehat{w} = 0.5$.

Let y be the upper right spel, with $\widehat{w} = 0.8$.

The optimal path is $p_o = \langle .5; .6; .5; .8 \rangle$,

giving $\hat{\rho}(x, y) = .3$

However, the Natural Algorithm returns

the path $p = \langle .5; .41; .5; .8 \rangle$

with $c_b(p) = .39 > \hat{\rho}(x, y)$.

Question (We do not know the answer)

Do the numbers $U(y) - L(y)$ returned by Natural Algorithm approximate $\hat{\rho}(x, y)$ in any reasonable sense?

So, how do we effectively compute the numbers $\hat{\rho}(x, y)$?

Failure of Natural Algorithm!

1	0.8
0.41	0.5
0.5	0.6

Let x be the lower left spel, with $\widehat{w} = 0.5$.

Let y be the upper right spel, with $\widehat{w} = 0.8$.

The optimal path is $p_o = \langle .5; .6; .5; .8 \rangle$,

giving $\hat{\rho}(x, y) = .3$

However, the Natural Algorithm returns

the path $p = \langle .5; .41; .5; .8 \rangle$

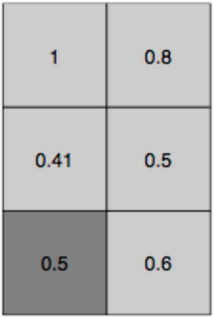
with $c_b(p) = .39 > \hat{\rho}(x, y)$.

Question (We do not know the answer)

Do the numbers $U(y) - L(y)$ returned by Natural Algorithm approximate $\hat{\rho}(x, y)$ in any reasonable sense?

So, how do we effectively compute the numbers $\hat{\rho}(x, y)$?

Failure of Natural Algorithm!



Let x be the lower left spel, with $\widehat{w} = 0.5$.

Let y be the upper right spel, with $\widehat{w} = 0.8$.

The optimal path is $p_o = \langle .5; .6; .5; .8 \rangle$,

giving $\hat{\rho}(x, y) = .3$

However, the Natural Algorithm returns

the path $p = \langle .5; .41; .5; .8 \rangle$

with $c_b(p) = .39 > \hat{\rho}(x, y)$.

Question (We do not know the answer)

Do the numbers $U(y) - L(y)$ returned by Natural Algorithm approximate $\hat{\rho}(x, y)$ in any reasonable sense?

So, how do we effectively compute the numbers $\hat{\rho}(x, y)$?

Failure of Natural Algorithm!

1	0.8
0.41	0.5
0.5	0.6

Let x be the lower left spel, with $\widehat{w} = 0.5$.

Let y be the upper right spel, with $\widehat{w} = 0.8$.

The optimal path is $p_o = \langle .5; .6; .5; .8 \rangle$,

giving $\hat{\rho}(x, y) = .3$

However, the Natural Algorithm returns

the path $p = \langle .5; .41; .5; .8 \rangle$

with $c_b(p) = .39 > \hat{\rho}(x, y)$.

Question (We do not know the answer)

Do the numbers $U(y) - L(y)$ returned by Natural Algorithm approximate $\hat{\rho}(x, y)$ in any reasonable sense?

So, how do we effectively compute the numbers $\hat{\rho}(x, y)$?

Better attempt to compute $\hat{\rho}(x, y)$

Let $\hat{\Pi}_{x,y} = \{p \colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$ and, for $p \in \hat{\Pi}_{x,y}$,

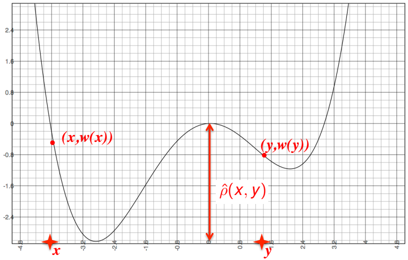
let $c_{\min}(p) = \min_t \widehat{w}(p(t))$ and $c_{\max}(p) = \max_t \widehat{w}(p(t))$.

Let $\hat{\varphi}(x, y) = \min_{p \in \hat{\Pi}_{x,y}} c_{\max}(p) - \max_{p \in \hat{\Pi}_{x,y}} c_{\min}(p)$

Clearly

$\hat{\varphi}(x, y) \leq \hat{\rho}(x, y)$

Is $\hat{\varphi}(x, y) = \hat{\rho}(x, y)$?



Better attempt to compute $\hat{\rho}(x, y)$

Let $\hat{\Pi}_{x,y} = \{p\colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$ and, for $p \in \hat{\Pi}_{x,y}$,

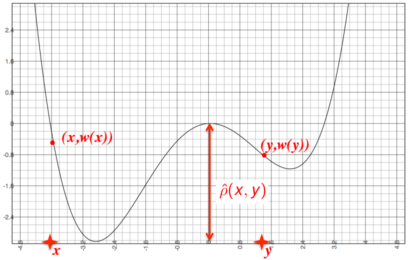
let $c_{\min}(p) = \min_t \widehat{w}(p(t))$ and $c_{\max}(p) = \max_t \widehat{w}(p(t))$.

Let $\hat{\varphi}(x, y) = \min_{p\in\hat{\Pi}_{x,y}} c_{\max}(p) - \max_{p\in\hat{\Pi}_{x,y}} c_{\min}(p)$

Clearly

$\hat{\varphi}(x, y) \leq \hat{\rho}(x, y)$

Is $\hat{\varphi}(x, y) = \hat{\rho}(x, y)$?



Better attempt to compute $\hat{\rho}(x, y)$

Let $\hat{\Pi}_{x,y} = \{p\colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$ and, for $p \in \hat{\Pi}_{x,y}$,

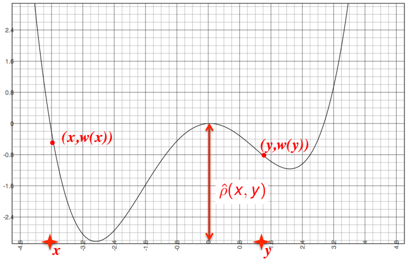
let $c_{\min}(p) = \min_t \widehat{w}(p(t))$ and $c_{\max}(p) = \max_t \widehat{w}(p(t))$.

Let $\hat{\varphi}(x, y) = \min_{p\in\hat{\Pi}_{x,y}} c_{\max}(p) - \max_{p\in\hat{\Pi}_{x,y}} c_{\min}(p)$

Clearly

$\hat{\varphi}(x, y) \leq \hat{\rho}(x, y)$

Is $\hat{\varphi}(x, y) = \hat{\rho}(x, y)$?



Better attempt to compute $\hat{\rho}(x, y)$

Let $\hat{\Pi}_{x,y} = \{p\colon p$ is a path in G from x to $y\}$ and, for $p \in \hat{\Pi}_{x,y}$,

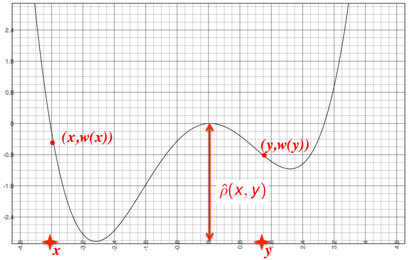
let $c_{\min}(p) = \min_t \widehat{w}(p(t))$ and $c_{\max}(p) = \max_t \widehat{w}(p(t))$.

Let $\hat{\varphi}(x, y) = \min_{p \in \hat{\Pi}_{x,y}} c_{\max}(p) - \max_{p \in \hat{\Pi}_{x,y}} c_{\min}(p)$

Clearly

$\hat{\varphi}(x, y) \leq \hat{\rho}(x, y)$

Is $\hat{\varphi}(x, y) = \hat{\rho}(x, y)$?



Better attempt to compute $\hat{\rho}(x, y)$

Let $\hat{\Pi}_{x,y} = \{p\colon p \text{ is a path in } G \text{ from } x \text{ to } y\}$ and, for $p \in \hat{\Pi}_{x,y}$,

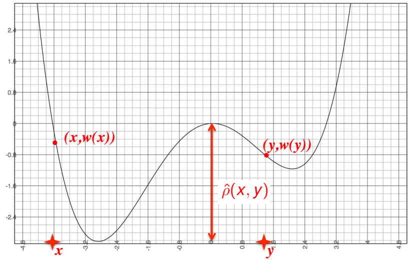
let $c_{\min}(p) = \min_t \widehat{w}(p(t))$ and $c_{\max}(p) = \max_t \widehat{w}(p(t))$.

Let $\hat{\varphi}(x, y) = \min_{p\in\hat{\Pi}_{x,y}} c_{\max}(p) - \max_{p\in\hat{\Pi}_{x,y}} c_{\min}(p)$

Clearly

$\hat{\varphi}(x, y) \leq \hat{\rho}(x, y)$

Is $\hat{\varphi}(x, y) = \hat{\rho}(x, y)$?



$\hat{\varphi}(x, y) \neq \hat{\rho}(x, y)$ but $\hat{\varphi}(x, y) \approx \hat{\rho}(x, y)$

y – upper right

0.41	0.5
0.5	0.6

x – lower left

$\min_{p \in \hat{\Pi}_{x,y}} c_{\max}(p) = c_{\max}(0.5, 0.41, 0.5) = 0.5$

$\max_{p \in \hat{\Pi}_{x,y}} c_{\min}(p) = c_{\min}(0.5, 0.6, 0.5) = 0.5$

But $\hat{\varphi}(x, y) = 0 \neq 0.09 = \hat{\rho}(x, y)$.

Nevertheless,

- $\hat{\varphi}(x, y) \approx \hat{\rho}(x, y)$, as we prove via continuous MBD.
- There is a very fast algorithm calculating $\hat{\varphi}(x, y)$.

$\hat{\varphi}(x, y) \neq \hat{\rho}(x, y)$ but $\hat{\varphi}(x, y) \approx \hat{\rho}(x, y)$

y – upper right

0.41	0.5
0.5	0.6

x – lower left

$\min_{p \in \hat{\Pi}_{x,y}} c_{\max}(p) = c_{\max}(0.5, 0.41, 0.5) = 0.5$

$\max_{p \in \hat{\Pi}_{x,y}} c_{\min}(p) = c_{\min}(0.5, 0.6, 0.5) = 0.5$

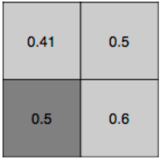
But $\hat{\varphi}(x, y) = 0 \neq 0.09 = \hat{\rho}(x, y)$.

Nevertheless,

- $\hat{\varphi}(x, y) \approx \hat{\rho}(x, y)$, as we prove via continuous MBD.
- There is a very fast algorithm calculating $\hat{\varphi}(x, y)$.

$\hat{\varphi}(x, y) \neq \hat{\rho}(x, y)$ but $\hat{\varphi}(x, y) \approx \hat{\rho}(x, y)$

y – upper right



x – lower left

$$\min_{p \in \hat{\Pi}_{x,y}} c_{\max}(p) = c_{\max}(0.5, 0.41, 0.5) = 0.5$$

$$\max_{p \in \hat{\Pi}_{x,y}} c_{\min}(p) = c_{\min}(0.5, 0.6, 0.5) = 0.5$$

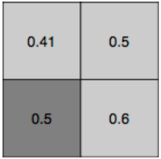
But $\hat{\varphi}(x, y) = 0 \neq 0.09 = \hat{\rho}(x, y)$.

Nevertheless,

- $\hat{\varphi}(x, y) \approx \hat{\rho}(x, y)$, as we prove via continuous MBD.
- There is a very fast algorithm calculating $\hat{\varphi}(x, y)$.

$\hat{\varphi}(x,y) \neq \hat{\rho}(x,y)$ but $\hat{\varphi}(x,y) \approx \hat{\rho}(x,y)$

y – upper right



x – lower left

$\min_{p \in \hat{\Pi}_{x,y}} c_{\max}(p) = c_{\max}(0.5, 0.41, 0.5) = 0.5$

$\max_{p \in \hat{\Pi}_{x,y}} c_{\min}(p) = c_{\min}(0.5, 0.6, 0.5) = 0.5$

But $\hat{\varphi}(x,y) = 0 \neq 0.09 = \hat{\rho}(x,y)$.

Nevertheless,

- $\hat{\varphi}(x,y) \approx \hat{\rho}(x,y)$, as we prove via continuous MBD.
- There is a very fast algorithm calculating $\hat{\varphi}(x,y)$.

$\hat{\varphi}(x,y) \neq \hat{\rho}(x,y)$ but $\hat{\varphi}(x,y) \approx \hat{\rho}(x,y)$

y – upper right

0.41	0.5
0.5	0.6

x – lower left

$$\min_{p \in \hat{\Pi}_{x,y}} c_{\max}(p) = c_{\max}(0.5, 0.41, 0.5) = 0.5$$

$$\max_{p \in \hat{\Pi}_{x,y}} c_{\min}(p) = c_{\min}(0.5, 0.6, 0.5) = 0.5$$

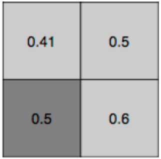
But $\hat{\varphi}(x,y) = 0 \neq 0.09 = \hat{\rho}(x,y)$.

Nevertheless,

- $\hat{\varphi}(x,y) \approx \hat{\rho}(x,y)$, as we prove via continuous MBD.
- There is a very fast algorithm calculating $\hat{\varphi}(x,y)$.

$\hat{\varphi}(x, y) \neq \hat{\rho}(x, y)$ but $\hat{\varphi}(x, y) \approx \hat{\rho}(x, y)$

y – upper right



x – lower left

$\min_{p \in \hat{\Pi}_{x,y}} c_{\max}(p) = c_{\max}(0.5, 0.41, 0.5) = 0.5$

$\max_{p \in \hat{\Pi}_{x,y}} c_{\min}(p) = c_{\min}(0.5, 0.6, 0.5) = 0.5$

But $\hat{\varphi}(x, y) = 0 \neq 0.09 = \hat{\rho}(x, y)$.

Nevertheless,

- $\hat{\varphi}(x, y) \approx \hat{\rho}(x, y)$, as we prove via continuous MBD.
- There is a very fast algorithm calculating $\hat{\varphi}(x, y)$.

Fast algorithm $A_{\hat{\varphi}}$ calculating $\hat{\varphi}(x,\cdot)$

Let $DA(G,\widehat{w},x)$ – Dijkstra algorithm returning p_y's with

$\hat{\rho}_{\max}(x,y) = c_{\max}(p_y)$.

Then $DA(G,-\widehat{w},x)$ returns p_y's with $\hat{\rho}_{\min}(x,y) = -c_{\max}(p_y)$.

Algorithm $A_{\hat{\varphi}}$:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: A map $\hat{\varphi}(x,\cdot)$.

1: Run $DA(G,\widehat{w},x)$ and record $C^{+}(y) = c_{\max}(p_y)$ for $y \in \widehat{D}$;
2: Run $DA(G,-\widehat{w},x)$ & record $C^{-}(y) = -c_{\max}(p_y)$ for $y \in \widehat{D}$;
3: Return $\hat{\varphi}(x,y) = C^{+}(y) - C^{-}(y)$ for every $y \in \widehat{D}$;

Algorithm $A_{\hat{\varphi}}$ requires $O(n \ln n)$ operations, n – the size of $\widehat{D}$.

Fast algorithm $A_{\hat{\varphi}}$ calculating $\hat{\varphi}(x, \cdot)$

Let $DA(G, \widehat{w}, x)$ – Dijkstra algorithm returning p_y's with

$\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Then $DA(G, -\widehat{w}, x)$ returns p_y's with $\hat{\rho}_{\min}(x, y) = -c_{\max}(p_y)$.

Algorithm $A_{\hat{\varphi}}$:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: A map $\hat{\varphi}(x, \cdot)$.

1: Run $DA(G, \widehat{w}, x)$ and record $C^+(y) = c_{\max}(p_y)$ for $y \in \widehat{D}$;

2: Run $DA(G, -\widehat{w}, x)$ & record $C^-(y) = -c_{\max}(p_y)$ for $y \in \widehat{D}$;

3: Return $\hat{\varphi}(x, y) = C^+(y) - C^-(y)$ for every $y \in \widehat{D}$;

Algorithm $A_{\hat{\varphi}}$ requires $O(n \ln n)$ operations, n – the size of $\widehat{D}$.

Fast algorithm $A_{\hat{\varphi}}$ calculating $\hat{\varphi}(x, \cdot)$

Let $DA(G, \widehat{w}, x)$ – Dijkstra algorithm returning p_y's with

$\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Then $DA(G, -\widehat{w}, x)$ returns p_y's with $\hat{\rho}_{\min}(x, y) = -c_{\max}(p_y)$.

Algorithm $A_{\hat{\varphi}}$:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: A map $\hat{\varphi}(x, \cdot)$.

1: Run $DA(G, \widehat{w}, x)$ and record $C^+(y) = c_{\max}(p_y)$ for $y \in \widehat{D}$;

2: Run $DA(G, -\widehat{w}, x)$ & record $C^-(y) = -c_{\max}(p_y)$ for $y \in \widehat{D}$;

3: Return $\hat{\varphi}(x, y) = C^+(y) - C^-(y)$ for every $y \in \widehat{D}$;

Algorithm $A_{\hat{\varphi}}$ requires $O(n \ln n)$ operations, n – the size of $\widehat{D}$.

Fast algorithm $A_{\hat{\varphi}}$ calculating $\hat{\varphi}(x,\cdot)$

Let $DA(G, \widehat{w}, x)$ – Dijkstra algorithm returning p_y's with

$\hat{\rho}_{\max}(x,y) = c_{\max}(p_y)$.

Then $DA(G, -\widehat{w}, x)$ returns p_y's with $\hat{\rho}_{\min}(x,y) = -c_{\max}(p_y)$.

Algorithm $A_{\hat{\varphi}}$:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: A map $\hat{\varphi}(x,\cdot)$.

1: Run $DA(G, \widehat{w}, x)$ and record $C^{+}(y) = c_{\max}(p_y)$ for $y \in \widehat{D}$;

2: Run $DA(G, -\widehat{w}, x)$ & record $C^{-}(y) = -c_{\max}(p_y)$ for $y \in \widehat{D}$;

3: Return $\hat{\varphi}(x,y) = C^{+}(y) - C^{-}(y)$ for every $y \in \widehat{D}$;

Algorithm $A_{\hat{\varphi}}$ requires $O(n \ln n)$ operations, n – the size of $\widehat{D}$.

Fast algorithm $A_{\hat{\varphi}}$ calculating $\hat{\varphi}(x, \cdot)$

Let $DA(G, \widehat{w}, x)$ – Dijkstra algorithm returning p_y's with

$\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Then $DA(G, -\widehat{w}, x)$ returns p_y's with $\hat{\rho}_{\min}(x, y) = -c_{\max}(p_y)$.

Algorithm $A_{\hat{\varphi}}$:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: A map $\hat{\varphi}(x, \cdot)$.

1: Run $DA(G, \widehat{w}, x)$ and record $C^+(y) = c_{\max}(p_y)$ for $y \in \widehat{D}$;

2: Run $DA(G, -\widehat{w}, x)$ & record $C^-(y) = -c_{\max}(p_y)$ for $y \in \widehat{D}$;

3: Return $\hat{\varphi}(x, y) = C^+(y) - C^-(y)$ for every $y \in \widehat{D}$;

Algorithm $A_{\hat{\varphi}}$ requires $O(n \ln n)$ operations, n – the size of $\widehat{D}$.

Fast algorithm $A_{\hat{\varphi}}$ calculating $\hat{\varphi}(x, \cdot)$

Let $DA(G, \widehat{w}, x)$ – Dijkstra algorithm returning p_y's with

$\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Then $DA(G, -\widehat{w}, x)$ returns p_y's with $\hat{\rho}_{\min}(x, y) = -c_{\max}(p_y)$.

Algorithm $A_{\hat{\varphi}}$:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: A map $\hat{\varphi}(x, \cdot)$.

1: Run $DA(G, \widehat{w}, x)$ and record $C^+(y) = c_{\max}(p_y)$ for $y \in \widehat{D}$;

2: Run $DA(G, -\widehat{w}, x)$ & record $C^-(y) = -c_{\max}(p_y)$ for $y \in \widehat{D}$;

3: Return $\hat{\varphi}(x, y) = C^+(y) - C^-(y)$ for every $y \in \widehat{D}$;

Algorithm $A_{\hat{\varphi}}$ requires $O(n \ln n)$ operations, n – the size of $\widehat{D}$.

Fast algorithm $A_{\hat{\varphi}}$ calculating $\hat{\varphi}(x, \cdot)$

Let $DA(G, \widehat{w}, x)$ – Dijkstra algorithm returning p_y's with

$\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Then $DA(G, -\widehat{w}, x)$ returns p_y's with $\hat{\rho}_{\min}(x, y) = -c_{\max}(p_y)$.

Algorithm $A_{\hat{\varphi}}$:

Input: A seed x in the image/graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$.

Output: A map $\hat{\varphi}(x, \cdot)$.

1: Run $DA(G, \widehat{w}, x)$ and record $C^+(y) = c_{\max}(p_y)$ for $y \in \widehat{D}$;

2: Run $DA(G, -\widehat{w}, x)$ & record $C^-(y) = -c_{\max}(p_y)$ for $y \in \widehat{D}$;

3: Return $\hat{\varphi}(x, y) = C^+(y) - C^-(y)$ for every $y \in \widehat{D}$;

Algorithm $A_{\hat{\varphi}}$ requires $O(n \ln n)$ operations, n – the size of $\widehat{D}$.

Outline

1. Minimum Barrier Distance, $\hat{\rho}$, in the discrete setting
2. How to compute $\hat{\rho}$?
3. Minimum Barrier Distance, ρ, in the continuous setting
4. Experiments: comparison with other distance measures
5. Newest result: fast algorithm for computing $\hat{\rho}$

Image, barrier cost of a path, and barrier distance

Input: Continuous function $f\colon D \to \mathbb{R}$, considered as an image,

where $D = \prod_{i=1}^{k} [a_i, b_i]$ $(a_i, b_i \in \mathbb{R})$.

For a (continuous) path $p\colon [0,1] \to D$ its barrier cost is

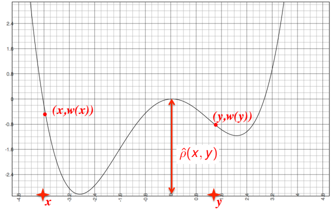
$$c_b(p) = \max_t w(p(t)) - \min_t w(p(t)), \quad \text{here } w = f.$$

(Note that max and min are attained, as $w \circ p$ is continuous.)

The continuous barrier distance

between $x, y \in D$ is given by:

$$\rho(x,y) = \inf\{c_b(p) \colon p \text{ from } x \text{ to } y\}$$



Image, barrier cost of a path, and barrier distance

Input: Continuous function $f\colon D \to \mathbb{R}$, considered as an image,

where $D = \prod_{i=1}^{k} [a_i, b_i]$ $(a_i, b_i \in \mathbb{R})$.

For a (continuous) path $p\colon [0,1] \to D$ its barrier cost is

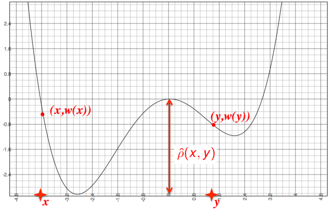
$$c_b(p) = \max_t w(p(t)) - \min_t w(p(t)), \quad \text{here } w = f.$$

(Note that max and min are attained, as $w \circ p$ is continuous.)

The continuous barrier distance

between $x, y \in D$ is given by:

$$\rho(x,y) = \inf\{c_b(p) : p \text{ from } x \text{ to } y\}$$



Image, barrier cost of a path, and barrier distance

Input: Continuous function $f\colon D \to \mathbb{R}$, considered as an image,

where $D = \prod_{i=1}^{k} [a_i, b_i]$ $(a_i, b_i \in \mathbb{R})$.

For a (continuous) path $p\colon [0,1] \to D$ its barrier cost is

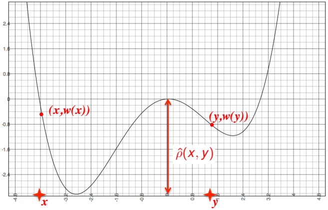
$$c_b(p) = \max_t w(p(t)) - \min_t w(p(t)), \quad \text{here } w = f.$$

(Note that max and min are attained, as $w \circ p$ is continuous.)

The continuous barrier distance

between $x, y \in D$ is given by:

$$\rho(x,y) = \inf\{c_b(p)\colon p \text{ from } x \text{ to } y\}$$



Image, barrier cost of a path, and barrier distance

Input: Continuous function $f\colon D \to \mathbb{R}$, considered as an image,

where $D = \prod_{i=1}^{k} [a_i, b_i]$ $(a_i, b_i \in \mathbb{R})$.

For a (continuous) path $p\colon [0,1] \to D$ its barrier cost is

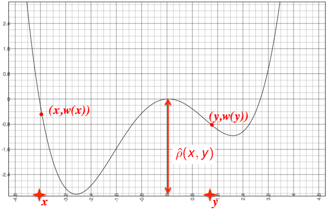
$$c_b(p) = \max_t w(p(t)) - \min_t w(p(t)), \quad \text{here } w = f.$$

(Note that max and min are attained, as $w \circ p$ is continuous.)

The continuous barrier distance

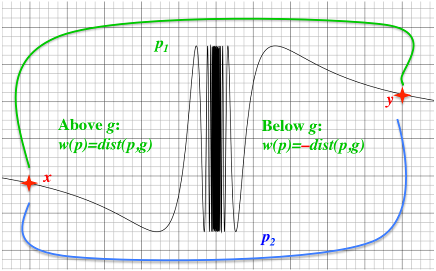
between $x, y \in D$ is given by:

$$\rho(x,y) = \inf\{c_b(p)\colon p \text{ from } x \text{ to } y\}$$



Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) : p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

For $\varphi(x, y) = \min_{p\in\Pi_{x,y}} c_{\max}(p) - \max_{p\in\Pi_{x,y}} c_{\min}(p)$

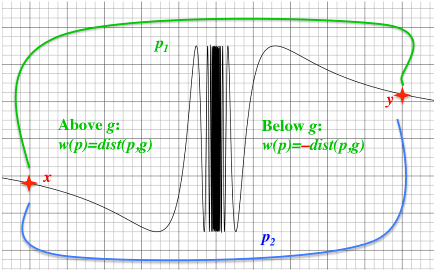
$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$

for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) : p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

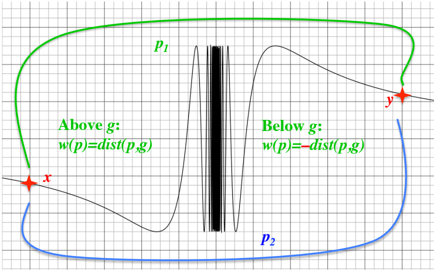
For $\varphi(x, y) = \min_{p \in \Pi_{x,y}} c_{\max}(p) - \max_{p \in \Pi_{x,y}} c_{\min}(p)$

$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$ for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) \colon p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

For $\varphi(x, y) = \min_{p \in \Pi_{x,y}} c_{\max}(p) - \max_{p \in \Pi_{x,y}} c_{\min}(p)$

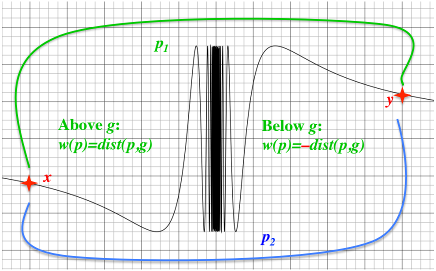
$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$

for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) \colon p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

For $\varphi(x, y) = \min_{p \in \Pi_{x,y}} c_{\max}(p) - \max_{p \in \Pi_{x,y}} c_{\min}(p)$

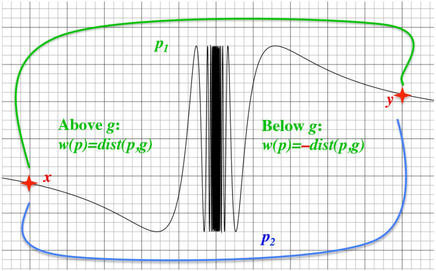
$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$

for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) \colon p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

For $\varphi(x, y) = \min_{p\in\Pi_{x,y}} c_{\max}(p) - \max_{p\in\Pi_{x,y}} c_{\min}(p)$

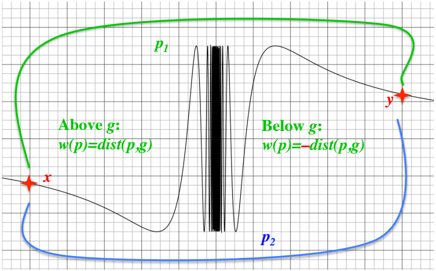
$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$

for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) \colon p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

For $\varphi(x, y) = \min_{p \in \Pi_{x,y}} c_{\max}(p) - \max_{p \in \Pi_{x,y}} c_{\min}(p)$

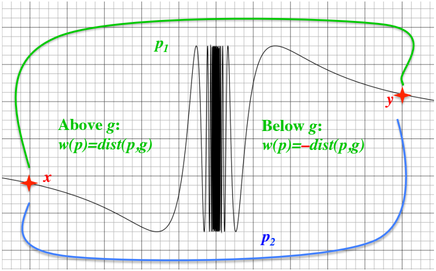
$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$

for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) \colon p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

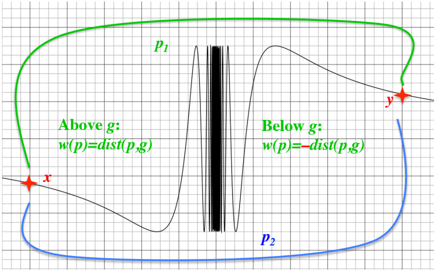
For $\varphi(x, y) = \min_{p \in \Pi_{x,y}} c_{\max}(p) - \max_{p \in \Pi_{x,y}} c_{\min}(p)$

$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$
for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) \colon p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

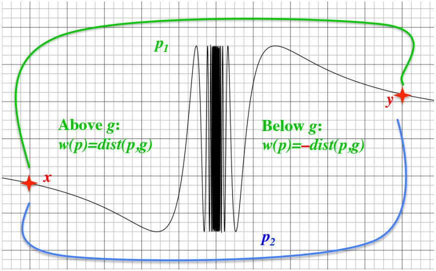
For $\varphi(x, y) = \min_{p \in \Pi_{x,y}} c_{\max}(p) - \max_{p \in \Pi_{x,y}} c_{\min}(p)$

$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$ for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) \colon p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

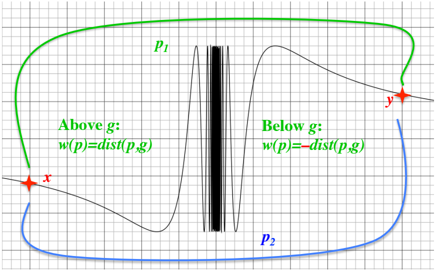
For $\varphi(x, y) = \min_{p \in \Pi_{x,y}} c_{\max}(p) - \max_{p \in \Pi_{x,y}} c_{\min}(p)$

$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$ for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Topologists sine curve example

$g(t) = \sin(1/t)$ for $t \neq 0$, $g(0) = 0$



$\rho(x, y) = \inf\{c_b(p) \colon p \in \Pi_{x,y}\}$

Put $c_{\min}(p) = \min_t w(p(t))$

and $c_{\max}(p) = \max_t w(p(t))$

$c_{\min}(p_1) = 0 < c_{\max}(p_1)$

$c_{\max}(p_2) = 0 > c_{\min}(p_2)$

For $\varphi(x, y) = \min_{p \in \Pi_{x,y}} c_{\max}(p) - \max_{p \in \Pi_{x,y}} c_{\min}(p)$

$c_{\max}(p_2) - c_{\min}(p_1) = 0 = \varphi(x, y) = \rho(x, y) < c_b(p)$

for any $p \in \Pi_{x,y}$.

In $\rho(x, y)$, operation inf cannot be replaced with min !

Continuous barrier distance to the rescue of $A_{\hat{\varphi}}$

Theorem (Deep result on simple connected domains)

If there are $p_1, p_2 \in \Pi_{x,y}$ with $a < c_{\min}(p_1)$ and $c_{\max}(p_2) < b$, then there is a single $p \in \Pi_{x,y}$ with the range in (a, b).

Corollary (continuous case)

$\varphi(x, y) = \rho(x, y)$ *for a w on a simple connected domain D.*

Theorem ($\hat{\varphi}(x, y) \to \varphi(x, y) = \rho(x, y)$ when $\hat{w} \to w$)

For every $x, y \in \widehat{D}$ there is a $p \in \hat{\Pi}_{x,y}$ with the range in the interval $[\hat{\rho}_{\min}(x, y) - \varepsilon, \hat{\rho}_{\max}(x, y) + \varepsilon]$, where
$\varepsilon = \max\{|w(x) - w(y)| : x, y \in \widehat{D}\ \&\ \max_i |x(i) - y(i)| \leq 1\}$.
In particular, $|\hat{\varphi}(x, y) - \hat{\rho}(x, y)| \leq 2\varepsilon$, $\hat{\varphi}(x, \cdot)$ returned by $A_{\hat{\varphi}}$.

Proof: (1) Extend $\hat{w}$ to w via k-linear interpolation. (2) Find p for w with $c_b(p) \approx \varphi(x, y) = \rho(x, y)$. (3) Digitize p.

Continuous barrier distance to the rescue of $A_{\hat{\varphi}}$

Theorem (Deep result on simple connected domains)

If there are $p_1, p_2 \in \Pi_{x,y}$ with $a < c_{\min}(p_1)$ and $c_{\max}(p_2) < b$, then there is a single $p \in \Pi_{x,y}$ with the range in (a, b).

Corollary (continuous case)

$\varphi(x,y) = \rho(x,y)$ *for a w on a simple connected domain D.*

Theorem ($\hat{\varphi}(x,y) \to \varphi(x,y) = \rho(x,y)$ when $\hat{w} \to w$)

For every $x, y \in \widehat{D}$ there is a $p \in \hat{\Pi}_{x,y}$ with the range in the interval $[\hat{\rho}_{\min}(x,y) - \varepsilon, \hat{\rho}_{\max}(x,y) + \varepsilon]$, where $\varepsilon = \max\{|w(x) - w(y)| : x, y \in \widehat{D} \;\&\; \max_i |x(i) - y(i)| \le 1\}$. In particular, $|\hat{\varphi}(x,y) - \hat{\rho}(x,y)| \le 2\varepsilon$, $\hat{\varphi}(x,\cdot)$ returned by $A_{\hat{\varphi}}$.

Proof: (1) Extend $\hat{w}$ to w via k-linear interpolation. (2) Find p for w with $c_b(p) \approx \varphi(x,y) = \rho(x,y)$. (3) Digitize p.

Continuous barrier distance to the rescue of $A_{\hat{\varphi}}$

Theorem (Deep result on simple connected domains)

If there are $p_1, p_2 \in \Pi_{x,y}$ with $a < c_{\min}(p_1)$ and $c_{\max}(p_2) < b$, then there is a single $p \in \Pi_{x,y}$ with the range in (a, b).

Corollary (continuous case)

$\varphi(x, y) = \rho(x, y)$ *for a w on a simple connected domain D.*

Theorem ($\hat{\varphi}(x, y) \to \varphi(x, y) = \rho(x, y)$ when $\hat{w} \to w$)

For every $x, y \in \widehat{D}$ there is a $p \in \hat{\Pi}_{x,y}$ with the range in the interval $[\hat{\rho}_{\min}(x, y) - \varepsilon, \hat{\rho}_{\max}(x, y) + \varepsilon]$, where
$\varepsilon = \max\{|w(x) - w(y)| : x, y \in \widehat{D}\ \&\ \max_i |x(i) - y(i)| \le 1\}$.
In particular, $|\hat{\varphi}(x, y) - \hat{\rho}(x, y)| \le 2\varepsilon$, $\hat{\varphi}(x, \cdot)$ returned by $A_{\hat{\varphi}}$.

Proof: (1) Extend $\hat{w}$ to w via k-linear interpolation. (2) Find p for w with $c_b(p) \approx \varphi(x, y) = \rho(x, y)$. (3) Digitize p.

Continuous barrier distance to the rescue of $A_{\hat{\varphi}}$

Theorem (Deep result on simple connected domains)

If there are $p_1, p_2 \in \Pi_{x,y}$ with $a < c_{\min}(p_1)$ and $c_{\max}(p_2) < b$, then there is a single $p \in \Pi_{x,y}$ with the range in (a, b).

Corollary (continuous case)

$\varphi(x, y) = \rho(x, y)$ *for a w on a simple connected domain D.*

Theorem ($\hat{\varphi}(x, y) \to \varphi(x, y) = \rho(x, y)$ when $\hat{w} \to w$)

For every $x, y \in \widehat{D}$ there is a $p \in \hat{\Pi}_{x,y}$ with the range in the interval $[\hat{\rho}_{\min}(x, y) - \varepsilon, \hat{\rho}_{\max}(x, y) + \varepsilon]$, where
$\varepsilon = \max\{|w(x) - w(y)| : x, y \in \widehat{D}\ \&\ \max_i |x(i) - y(i)| \le 1\}$.
In particular, $|\hat{\varphi}(x, y) - \hat{\rho}(x, y)| \le 2\varepsilon$, $\hat{\varphi}(x, \cdot)$ returned by $A_{\hat{\varphi}}$.

Proof: (1) Extend $\hat{w}$ to w via k-linear interpolation. (2) Find p for w with $c_b(p) \approx \varphi(x, y) = \rho(x, y)$. (3) Digitize p.

Continuous barrier distance to the rescue of $A_{\hat{\varphi}}$

Theorem (Deep result on simple connected domains)

If there are $p_1, p_2 \in \Pi_{x,y}$ with $a < c_{\min}(p_1)$ and $c_{\max}(p_2) < b$, then there is a single $p \in \Pi_{x,y}$ with the range in (a, b).

Corollary (continuous case)

$\varphi(x, y) = \rho(x, y)$ *for a w on a simple connected domain D.*

Theorem ($\hat{\varphi}(x, y) \to \varphi(x, y) = \rho(x, y)$ when $\hat{w} \to w$)

For every $x, y \in \widehat{D}$ there is a $p \in \hat{\Pi}_{x,y}$ with the range in the interval $[\hat{\rho}_{\min}(x, y) - \varepsilon, \hat{\rho}_{\max}(x, y) + \varepsilon]$, where
$\varepsilon = \max\{|w(x) - w(y)| : x, y \in \widehat{D} \ \&\ \max_i |x(i) - y(i)| \leq 1\}$.
In particular, $|\hat{\varphi}(x, y) - \hat{\rho}(x, y)| \leq 2\varepsilon$, $\hat{\varphi}(x, \cdot)$ returned by $A_{\hat{\varphi}}$.

Proof: (1) Extend $\hat{w}$ to w via k-linear interpolation. (2) Find p for w with $c_b(p) \approx \varphi(x, y) = \rho(x, y)$. (3) Digitize p.

Continuous barrier distance to the rescue of $A_{\hat{\varphi}}$

Theorem (Deep result on simple connected domains)

If there are $p_1, p_2 \in \Pi_{x,y}$ with $a < c_{\min}(p_1)$ and $c_{\max}(p_2) < b$, then there is a single $p \in \Pi_{x,y}$ with the range in (a, b).

Corollary (continuous case)

$\varphi(x, y) = \rho(x, y)$ for a w on a simple connected domain D.

Theorem ($\hat{\varphi}(x, y) \to \varphi(x, y) = \rho(x, y)$ when $\hat{w} \to w$)

For every $x, y \in \widehat{D}$ there is a $p \in \hat{\Pi}_{x,y}$ with the range in the interval $[\hat{\rho}_{\min}(x, y) - \varepsilon, \hat{\rho}_{\max}(x, y) + \varepsilon]$, where
$\varepsilon = \max\{|w(x) - w(y)| : x, y \in \widehat{D} \ \& \ \max_i |x(i) - y(i)| \leq 1\}$.
In particular, $|\hat{\varphi}(x, y) - \hat{\rho}(x, y)| \leq 2\varepsilon$, $\hat{\varphi}(x, \cdot)$ returned by $A_{\hat{\varphi}}$.

Proof: (1) Extend $\hat{w}$ to w via k-linear interpolation. (2) Find p for w with $c_b(p) \approx \varphi(x, y) = \rho(x, y)$. (3) Digitize p.

Continuous barrier distance to the rescue of $A_{\hat{\varphi}}$

Theorem (Deep result on simple connected domains)

If there are $p_1, p_2 \in \Pi_{x,y}$ *with* $a < c_{\min}(p_1)$ *and* $c_{\max}(p_2) < b$, *then there is a single* $p \in \Pi_{x,y}$ *with the range in* (a, b).

Corollary (continuous case)

$\varphi(x, y) = \rho(x, y)$ *for a* w *on a simple connected domain* D.

Theorem ($\hat{\varphi}(x, y) \to \varphi(x, y) = \rho(x, y)$ when $\hat{w} \to w$)

For every $x, y \in \widehat{D}$ *there is a* $p \in \hat{\Pi}_{x,y}$ *with the range in the interval* $[\hat{\rho}_{\min}(x, y) - \varepsilon, \hat{\rho}_{\max}(x, y) + \varepsilon]$, *where*
$\varepsilon = \max\{|w(x) - w(y)| : x, y \in \widehat{D}\ \&\ \max_i |x(i) - y(i)| \le 1\}$.
In particular, $|\hat{\varphi}(x, y) - \hat{\rho}(x, y)| \le 2\varepsilon$, $\hat{\varphi}(x, \cdot)$ *returned by* $A_{\hat{\varphi}}$.

Proof: (1) Extend $\hat{w}$ to w via k-linear interpolation. (2) Find p for w with $c_b(p) \approx \varphi(x, y) = \rho(x, y)$. (3) Digitize p.

Continuous barrier distance to the rescue of $A_{\hat{\varphi}}$

Theorem (Deep result on simple connected domains)

If there are $p_1, p_2 \in \Pi_{x,y}$ with $a < c_{\min}(p_1)$ and $c_{\max}(p_2) < b$, then there is a single $p \in \Pi_{x,y}$ with the range in (a, b).

Corollary (continuous case)

$\varphi(x, y) = \rho(x, y)$ *for a w on a simple connected domain D.*

Theorem ($\hat{\varphi}(x, y) \to \varphi(x, y) = \rho(x, y)$ when $\hat{w} \to w$)

For every $x, y \in \widehat{D}$ *there is a* $p \in \hat{\Pi}_{x,y}$ *with the range in the interval* $[\hat{\rho}_{\min}(x, y) - \varepsilon, \hat{\rho}_{\max}(x, y) + \varepsilon]$, *where*
$\varepsilon = \max\{|w(x) - w(y)| : x, y \in \widehat{D}\ \&\ \max_i |x(i) - y(i)| \leq 1\}$.
In particular, $|\hat{\varphi}(x, y) - \hat{\rho}(x, y)| \leq 2\varepsilon$, $\hat{\varphi}(x, \cdot)$ *returned by* $A_{\hat{\varphi}}$.

Proof: (1) Extend $\hat{w}$ to w via k-linear interpolation. (2) Find p for w with $c_b(p) \approx \varphi(x, y) = \rho(x, y)$. (3) Digitize p.

Outline

1. Minimum Barrier Distance, $\hat{\rho}$, in the discrete setting
2. How to compute $\hat{\rho}$?
3. Minimum Barrier Distance, ρ, in the continuous setting
4. Experiments: comparison with other distance measures
5. Newest result: fast algorithm for computing $\hat{\rho}$

Experiments

We compared the output $\hat{\varphi}(x, y)$ of $A_{\hat{\varphi}}$ (approximating $\hat{\rho}(x, y)$) with the distances minimizing costs $c(p)$, $p = \langle p_1, p_2, \ldots, p_m \rangle$:

- fuzzy d_F; $c(p) = \sum_{i=1}^{m-1} \frac{f_A(p_i)+f_A(p_{i+1})}{2} \cdot \|p_i - p_{i+1}\|$;
- geodesic d_G; $c(p) = \sum_i \omega \, |f_A(p_i) - f_A(p_{i+1})| + \|p_i - p_{i+1}\|$;

FC: max-arc $d_{\max}$; $c(p) = \max_{i=1,\ldots,m-1} |f_A(p_i) - f_A(p_{i+1})|$;

We compared the distances with respect to:

(A) the ratios between inter-object and intra-object distances & the influence by the seed points position: should be low;

(B) influence by introduction of noise & smoothing (blur): should be small.

Experiments

We compared the output $\hat{\varphi}(x, y)$ of $A_{\hat{\varphi}}$ (approximating $\hat{\rho}(x, y)$) with the distances minimizing costs $c(p)$, $p = \langle p_1, p_2, \ldots, p_m \rangle$:

- fuzzy d_F; $c(p) = \sum_{i=1}^{m-1} \frac{f_A(p_i)+f_A(p_{i+1})}{2} \cdot \|p_i - p_{i+1}\|$;
- geodesic d_G; $c(p) = \sum_i \omega \left|f_A(p_i) - f_A(p_{i+1})\right| + \|p_i - p_{i+1}\|$;

FC: max-arc $d_{\max}$; $c(p) = \max_{i=1,\ldots,m-1} \left|f_A(p_i) - f_A(p_{i+1})\right|$;

We compared the distances with respect to:

(A) the ratios between inter-object and intra-object distances & the influence by the seed points position: should be low;

(B) influence by introduction of noise & smoothing (blur): should be small.

Experiments

We compared the output $\hat{\varphi}(x, y)$ of $A_{\hat{\varphi}}$ (approximating $\hat{\rho}(x, y)$) with the distances minimizing costs $c(p)$, $p = \langle p_1, p_2, \ldots, p_m \rangle$:

- fuzzy d_F; $c(p) = \sum_{i=1}^{m-1} \frac{f_A(p_i)+f_A(p_{i+1})}{2} \cdot \|p_i - p_{i+1}\|$;
- geodesic d_G; $c(p) = \sum_i \omega \, |f_A(p_i) - f_A(p_{i+1})| + \|p_i - p_{i+1}\|$;

FC: max-arc $d_{\max}$; $c(p) = \max_{i=1,\ldots,m-1} |f_A(p_i) - f_A(p_{i+1})|$;

We compared the distances with respect to:

(A) the ratios between inter-object and intra-object distances & the influence by the seed points position: should be low;

(B) influence by introduction of noise & smoothing (blur): should be small.

Experiments

We compared the output $\hat{\varphi}(x,y)$ of $A_{\hat{\varphi}}$ (approximating $\hat{\rho}(x,y)$) with the distances minimizing costs $c(p)$, $p = \langle p_1, p_2, \ldots, p_m \rangle$:

- fuzzy d_F; $c(p) = \sum_{i=1}^{m-1} \frac{f_A(p_i)+f_A(p_{i+1})}{2} \cdot \|p_i - p_{i+1}\|$;
- geodesic d_G; $c(p) = \sum_i \omega \, |f_A(p_i) - f_A(p_{i+1})| + \|p_i - p_{i+1}\|$;

FC: max-arc $d_{\max}$; $c(p) = \max_{i=1,\ldots,m-1} |f_A(p_i) - f_A(p_{i+1})|$;

We compared the distances with respect to:

(A) the ratios between inter-object and intra-object distances & the influence by the seed points position: should be low;

(B) influence by introduction of noise & smoothing (blur): should be small.

Experiments

We compared the output $\hat{\varphi}(x,y)$ of $A_{\hat{\varphi}}$ (approximating $\hat{\rho}(x,y)$) with the distances minimizing costs $c(p)$, $p = \langle p_1, p_2, \ldots, p_m \rangle$:

- fuzzy d_F; $c(p) = \sum_{i=1}^{m-1} \frac{f_A(p_i)+f_A(p_{i+1})}{2} \cdot \|p_i - p_{i+1}\|$;
- geodesic d_G; $c(p) = \sum_i \omega \, |f_A(p_i) - f_A(p_{i+1})| + \|p_i - p_{i+1}\|$;

FC: max-arc $d_{\max}$; $c(p) = \max_{i=1,\ldots,m-1} |f_A(p_i) - f_A(p_{i+1})|$;

We compared the distances with respect to:

(A) the ratios between inter-object and intra-object distances & the influence by the seed points position: should be low;

(B) influence by introduction of noise & smoothing (blur): should be small.

Experiments

We compared the output $\hat{\varphi}(x, y)$ of $A_{\hat{\varphi}}$ (approximating $\hat{\rho}(x, y)$) with the distances minimizing costs $c(p)$, $p = \langle p_1, p_2, \ldots, p_m \rangle$:

- fuzzy d_F; $c(p) = \sum_{i=1}^{m-1} \frac{f_A(p_i)+f_A(p_{i+1})}{2} \cdot \|p_i - p_{i+1}\|$;
- geodesic d_G; $c(p) = \sum_i \omega \, |f_A(p_i) - f_A(p_{i+1})| + \|p_i - p_{i+1}\|$;

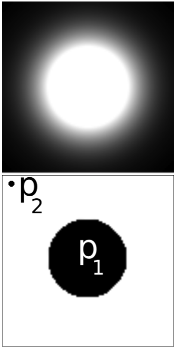
FC: max-arc $d_{\max}$; $c(p) = \max_{i=1,\ldots,m-1} |f_A(p_i) - f_A(p_{i+1})|$;

We compared the distances with respect to:

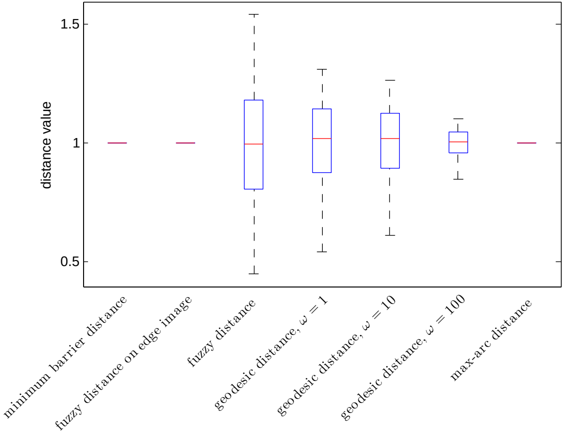
(A) the ratios between inter-object and intra-object distances & the influence by the seed points position: should be low;

(B) influence by introduction of noise & smoothing (blur): should be small.

Stability w.r.t. seed position

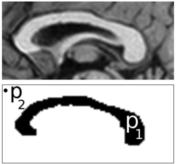


p_2 fixed;
p_1 is chosen randomly 1000 times

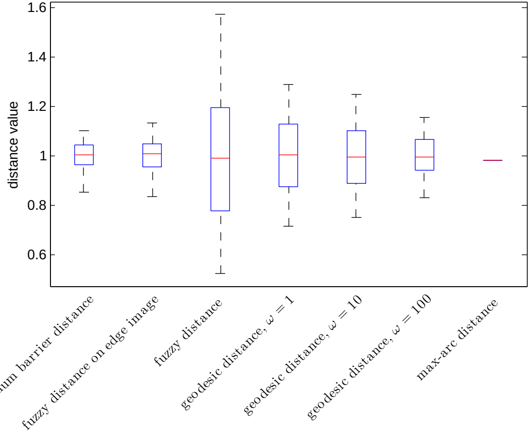


Boxes: 25th to the 75th percentile; central mark: the median.

Stability w.r.t. seed position



p_2 fixed;
p_1 is chosen randomly 1000 times



Boxes: 25th to the 75th percentile; central mark: the median.

Stability w.r.t. seed position: interpretation

- max-arc (Fuzzy Connectedness) distance is the most robust;
- MBD is just slightly worst than max-arc and only for the image with a low boundary gradient;
- MBD is at least as good than the other distances;

Stability w.r.t. seed position: interpretation

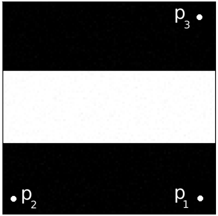
- max-arc (Fuzzy Connectedness) distance is the most robust;
- MBD is just slightly worst than max-arc and only for the image with a low boundary gradient;
- MBD is at least as good than the other distances;

Stability w.r.t. seed position: interpretation

- max-arc (Fuzzy Connectedness) distance is the most robust;
- MBD is just slightly worst than max-arc and only for the image with a low boundary gradient;
- MBD is at least as good than the other distances;

Stability w.r.t. Gaussian noise and smoothing

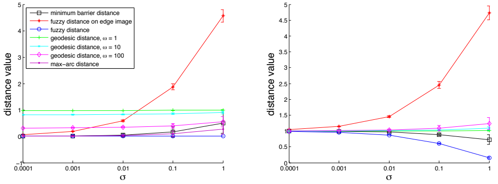
Test image



p_2 and p_3 are fixed

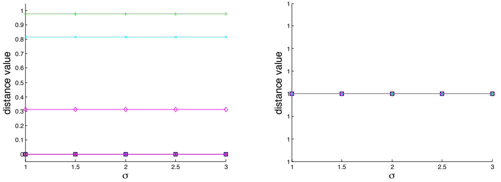
p_1 is randomly chosen 1000 times.

Gaussian noise – Distance values as function of sigma



Intra-object distance $(d(p_1, p_2))$ Inter-object distance $(d(p_1, p_3))$

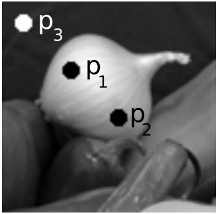
Gaussian smoothing – Distance values as function of sigma



Intra-object distance $(d(p_1, p_2))$ Inter-object distance $(d(p_1, p_3))$

Stability w.r.t. Gaussian noise and smoothing

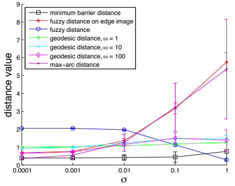
Test image



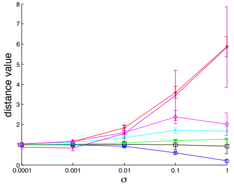
p_2 and p_3 are fixed

p_1 is randomly chosen 1000 times.

Gaussian noise – Distance values as function of sigma

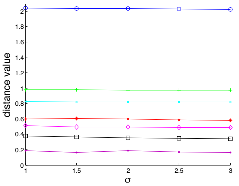


Intra-object distance $(d(p_1, p_2))$

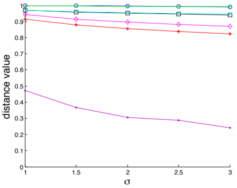


Inter-object distance $(d(p_1, p_3))$

Gaussian smoothing – Distance values as function of sigma



Intra-object distance $(d(p_1, p_2))$



Inter-object distance $(d(p_1, p_3))$

Interpretation:

Sensitivity to noise and blur:

- MBD has low sensitivity;
- fuzzy distance on edge image and max-arc are sensitive;
- fuzzy distance: performs well for the image with a high boundary gradient; not so well for the image with a low boundary gradient.

Separation of the object from the background (ratio):

- all considered distances perform reasonably well;
- the performance of max-arc (FC) decreases, with weakening boundary gradient and/or introduction of noise; no such decrease for MBD;

Interpretation:

Sensitivity to noise and blur:

- MBD has low sensitivity;
- fuzzy distance on edge image and max-arc are sensitive;
- fuzzy distance: performs well for the image with a high boundary gradient; not so well for the image with a low boundary gradient.

Separation of the object from the background (ratio):

- all considered distances perform reasonably well;
- the performance of max-arc (FC) decreases, with weakening boundary gradient and/or introduction of noise; no such decrease for MBD;

Interpretation:

Sensitivity to noise and blur:

- MBD has low sensitivity;
- fuzzy distance on edge image and max-arc are sensitive;
- fuzzy distance: performs well for the image with a high boundary gradient; not so well for the image with a low boundary gradient.

Separation of the object from the background (ratio):

- all considered distances perform reasonably well;
- the performance of max-arc (FC) decreases, with weakening boundary gradient and/or introduction of noise; no such decrease for MBD;

Interpretation:

Sensitivity to noise and blur:

- MBD has low sensitivity;
- fuzzy distance on edge image and max-arc are sensitive;
- fuzzy distance: performs well for the image with a high boundary gradient; not so well for the image with a low boundary gradient.

Separation of the object from the background (ratio):

- all considered distances perform reasonably well;
- the performance of max-arc (FC) decreases, with weakening boundary gradient and/or introduction of noise; no such decrease for MBD;

Interpretation:

Sensitivity to noise and blur:

- MBD has low sensitivity;
- fuzzy distance on edge image and max-arc are sensitive;
- fuzzy distance: performs well for the image with a high boundary gradient; not so well for the image with a low boundary gradient.

Separation of the object from the background (ratio):

- all considered distances perform reasonably well;
- the performance of max-arc (FC) decreases, with weakening boundary gradient and/or introduction of noise; no such decrease for MBD;

Outline

1. Minimum Barrier Distance, $\hat{\rho}$, in the discrete setting
2. How to compute $\hat{\rho}$?
3. Minimum Barrier Distance, ρ, in the continuous setting
4. Experiments: comparison with other distance measures
5. Newest result: fast algorithm for computing $\hat{\rho}$

Newest fast algorithm for computing $\hat{\rho}$

$DA(G, \widehat{w}, x)$ – returns p_y's with $\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Define $w_a(x) = w(x)$ for $w(x) \geq a$ and $w_a(x) = \infty$ otherwise.

Algorithm $A_{\hat{\rho}}$:

Input: Graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$ and the vertex x in G.
Output: A path $\hat{p}_y$ in G from x to y with $c_b(\hat{p}_y) = \hat{\rho}(x, y)$.
Auxiliary: Current value C_b of $c_b(\hat{p}_y)$;

```
List R = {w(c) ≤ w(x): c ∈ D̂} with no repetition;
for every a ∈ R do
    Run DA(G, w_a, x);
    if c_b(p_y) < C_b then
        Put C_b = c_b(p_y) and p̂_y = p_y;
    end if
end for
Return p̂_y;
```

Newest fast algorithm for computing $\hat{\rho}$

$DA(G, \widehat{w}, x)$ – returns p_y's with $\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Define $w_a(x) = w(x)$ for $w(x) \geq a$ and $w_a(x) = \infty$ otherwise.

Algorithm $A_{\hat{\rho}}$:

Input: Graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$ and the vertex x in G.
Output: A path $\hat{p}_y$ in G from x to y with $c_b(\hat{p}_y) = \hat{\rho}(x, y)$.
Auxiliary: Current value C_b of $c_b(\hat{p}_y)$;

1: List $R = \{w(c) \leq w(x) \colon c \in \hat{D}\}$ with no repetition;
2: **for** every $a \in R$ **do**
3: Run $DA(G, w_a, x)$;
4: **if** $c_b(p_y) < C_b$ **then**
5: Put $C_b = c_b(p_y)$ and $\hat{p}_y = p_y$;
6: **end if**
7: **end for**
8: Return $\hat{p}_y$;

Newest fast algorithm for computing $\hat{\rho}$

$DA(G, \widehat{w}, x)$ – returns p_y's with $\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Define $w_a(x) = w(x)$ for $w(x) \geq a$ and $w_a(x) = \infty$ otherwise.

Algorithm $A_{\hat{\rho}}$:

Input: Graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$ and the vertex x in G.
Output: A path $\hat{p}_y$ in G from x to y with $c_b(\hat{p}_y) = \hat{\rho}(x, y)$.
Auxiliary: Current value C_b of $c_b(\hat{p}_y)$;

```
List R = {w(c) ≤ w(x): c ∈ D̂} with no repetition;
for every a ∈ R do
    Run DA(G, w_a, x);
    if c_b(p_y) < C_b then
        Put C_b = c_b(p_y) and p̂_y = p_y;
    end if
end for
Return p̂_y;
```

Newest fast algorithm for computing $\hat{\rho}$

$DA(G, \widehat{w}, x)$ – returns p_y's with $\hat{\rho}_{\max}(x, y) = c_{\max}(p_y)$.

Define $w_a(x) = w(x)$ for $w(x) \geq a$ and $w_a(x) = \infty$ otherwise.

Algorithm $A_{\hat{\rho}}$:

Input: Graph $G = \langle \widehat{D}, E, \widehat{w} \rangle$ and the vertex x in G.
Output: A path $\hat{p}_y$ in G from x to y with $c_b(\hat{p}_y) = \hat{\rho}(x, y)$.
Auxiliary: Current value C_b of $c_b(\hat{p}_y)$;

```
List R = {w(c) ≤ w(x): c ∈ D̂} with no repetition;
for every a ∈ R do
    Run DA(G, w_a, x);
    if c_b(p_y) < C_b then
        Put C_b = c_b(p_y) and p̂_y = p_y;
    end if
end for
Return p̂_y;
```

On the algorithm $A_{\hat{\rho}}$

Theorem (KC: Proved about two weeks ago)

$A_{\hat{\rho}}$ returns paths $\hat{p}_y$ with the exact values $c_b(\hat{p}_y) = \hat{\rho}(x, y)$.

$A_{\hat{\rho}}$ requires $O(k(n + k))$ operations
n – the size of $\widehat{D}$, and k – the size of $\{w(c) \leq w(x) \colon c \in \hat{D}\}$.

This estimate reduces to $O(n)$, when $k \ll n$,

usually true in the image processing.

On the algorithm $A_{\hat{\rho}}$

Theorem (KC: Proved about two weeks ago)

$A_{\hat{\rho}}$ returns paths $\hat{p}_y$ with the exact values $c_b(\hat{p}_y) = \hat{\rho}(x, y)$.

$A_{\hat{\rho}}$ requires $O(k(n + k))$ operations
n – the size of $\widehat{D}$, and k – the size of $\{w(c) \leq w(x) \colon c \in \hat{D}\}$.

This estimate reduces to $O(n)$, when $k \ll n$,

usually true in the image processing.

On the algorithm $A_{\hat{\rho}}$

Theorem (KC: Proved about two weeks ago)

$A_{\hat{\rho}}$ returns paths $\hat{p}_y$ with the exact values $c_b(\hat{p}_y) = \hat{\rho}(x, y)$.

$A_{\hat{\rho}}$ requires $O(k(n+k))$ operations
n – the size of $\widehat{D}$, and k – the size of $\{w(c) \leq w(x) \colon c \in \hat{D}\}$.

This estimate reduces to $O(n)$, when $k \ll n$,

usually true in the image processing.

Summary

Minimum Barrier Distance $\hat{\rho}$ — novel quasi distance function:

- Can be effectively computed.
- Is quite stable with respect to:
  change of seed position and introduction of noise or blur.
  (Comparing to fuzzy, geodesic, and max-arc distances.)
- $\hat{\rho}(x, y)$ measures:
  - homogeneity, for $|\hat{w}(x) - \hat{w}(y)|$ small;
  - $\approx |\hat{w}(x) - \hat{w}(y)|$ (c.a. object feature) for $|\hat{w}(x) - \hat{w}(y)|$ large.

More experimental evaluation of $\hat{\rho}$ is still needed.

Summary

Minimum Barrier Distance $\hat{\rho}$ — novel quasi distance function:

- Can be effectively computed.
- Is quite stable with respect to:
  change of seed position and introduction of noise or blur.
  (Comparing to fuzzy, geodesic, and max-arc distances.)
- $\hat{\rho}(x, y)$ measures:
  - homogeneity, for $|\hat{w}(x) - \hat{w}(y)|$ small;
  - $\approx |\hat{w}(x) - \hat{w}(y)|$ (c.a. object feature) for $|\hat{w}(x) - \hat{w}(y)|$ large.

More experimental evaluation of $\hat{\rho}$ is still needed.

Summary

Minimum Barrier Distance $\hat{\rho}$ — novel quasi distance function:

- Can be effectively computed.
- Is quite stable with respect to:
  change of seed position and introduction of noise or blur.
  (Comparing to fuzzy, geodesic, and max-arc distances.)
- $\hat{\rho}(x, y)$ measures:
  - homogeneity, for $|\hat{w}(x) - \hat{w}(y)|$ small;
  - $\approx |\hat{w}(x) - \hat{w}(y)|$ (c.a. object feature) for $|\hat{w}(x) - \hat{w}(y)|$ large.

More experimental evaluation of $\hat{\rho}$ is still needed.

Summary

Minimum Barrier Distance $\hat{\rho}$ — novel quasi distance function:

- Can be effectively computed.
- Is quite stable with respect to:
  change of seed position and introduction of noise or blur.
  (Comparing to fuzzy, geodesic, and max-arc distances.)
- $\hat{\rho}(x, y)$ measures:
  - homogeneity, for $|\hat{w}(x) - \hat{w}(y)|$ small;
  - $\approx |\hat{w}(x) - \hat{w}(y)|$ (c.a. object feature) for $|\hat{w}(x) - \hat{w}(y)|$ large.

More experimental evaluation of $\hat{\rho}$ is still needed.

Summary

Minimum Barrier Distance $\hat{\rho}$ — novel quasi distance function:

- Can be effectively computed.
- Is quite stable with respect to:
  change of seed position and introduction of noise or blur.
  (Comparing to fuzzy, geodesic, and max-arc distances.)
- $\hat{\rho}(x, y)$ measures:
  - homogeneity, for $|\hat{w}(x) - \hat{w}(y)|$ small;
  - $\approx |\hat{w}(x) - \hat{w}(y)|$ (c.a. object feature) for $|\hat{w}(x) - \hat{w}(y)|$ large.

More experimental evaluation of $\hat{\rho}$ is still needed.

Summary

Minimum Barrier Distance $\hat{\rho}$ — novel quasi distance function:

- Can be effectively computed.
- Is quite stable with respect to:
  change of seed position and introduction of noise or blur.
  (Comparing to fuzzy, geodesic, and max-arc distances.)
- $\hat{\rho}(x, y)$ measures:
  - homogeneity, for $|\hat{w}(x) - \hat{w}(y)|$ small;
  - $\approx |\hat{w}(x) - \hat{w}(y)|$ (c.a. object feature) for $|\hat{w}(x) - \hat{w}(y)|$ large.

More experimental evaluation of $\hat{\rho}$ is still needed.

Thank you for your attention!