Welcome back...

Welcome back...

Metric spaces.

A metric space X, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Metric spaces.

A metric space X, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) X from R^d and $d(\cdot,\cdot)$ is Euclidean distance.

(B) X from R^d and $d(\cdot,\cdot)$ is squared Euclidean distance.

(C) X- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) X is a set of vectors and $d(u,v)$ is $u \cdot v$.

Metric spaces.

A metric space X, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) X from R^d and $d(\cdot,\cdot)$ is Euclidean distance.

(B) X from R^d and $d(\cdot,\cdot)$ is squared Euclidean distance.

(C) X- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) X is a set of vectors and $d(u,v)$ is $u \cdot v$.

Metric spaces.

A metric space X, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) X from R^d and $d(\cdot,\cdot)$ is Euclidean distance.

(B) X from R^d and $d(\cdot,\cdot)$ is squared Euclidean distance.

(C) X- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) X is a set of vectors and $d(u,v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling.

Metric spaces.

A metric space X, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) X from R^d and $d(\cdot,\cdot)$ is Euclidean distance.

(B) X from R^d and $d(\cdot,\cdot)$ is squared Euclidean distance.

(C) X- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) X is a set of vectors and $d(u,v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling.

Hard problems.

Metric spaces.

A metric space X, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) X from R^d and $d(\cdot,\cdot)$ is Euclidean distance.

(B) X from R^d and $d(\cdot,\cdot)$ is squared Euclidean distance.

(C) X- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) X is a set of vectors and $d(u,v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling.

Hard problems. Easier to solve on trees.

Metric spaces.

A metric space X, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) X from R^d and $d(\cdot,\cdot)$ is Euclidean distance.

(B) X from R^d and $d(\cdot,\cdot)$ is squared Euclidean distance.

(C) X- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) X is a set of vectors and $d(u,v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling.

Hard problems. Easier to solve on trees. Dynamic programming on trees.

Metric spaces.

A metric space X, $d(i,j)$ where $d(i,j) \leq d(i,k) + d(k,j)$, $d(i,j) = d(j,i)$, and $d(i,j) \geq 0$.

Which are metric spaces?

(A) X from R^d and $d(\cdot,\cdot)$ is Euclidean distance.

(B) X from R^d and $d(\cdot,\cdot)$ is squared Euclidean distance.

(C) X- vertices in graph, $d(i,j)$ is shortest path distances in graph.

(D) X is a set of vectors and $d(u,v)$ is $u \cdot v$.

Input to TSP, facility location, some layout problems, ..., metric labelling.

Hard problems. Easier to solve on trees. Dynamic programming on trees.

Approximate metric on trees?

Approximate metric using a tree.

Tree metric:

Approximate metric using a tree.

Tree metric:

X is nodes of tree with edge weights

$d_T(i,j)$ shortest path metric on tree.

Approximate metric using a tree.

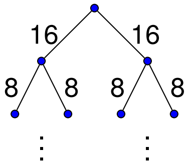
Tree metric:
X is nodes of tree with edge weights
$d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:

Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.

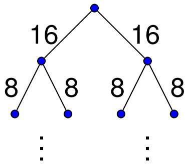
Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.



Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.



Map X into tree.

Approximate metric using a tree.

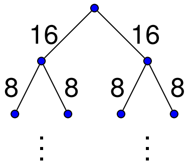
Tree metric:

  X is nodes of tree with edge weights

  $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:

  Tree weights are geometrically decreasing.



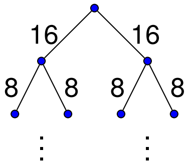
  Map X into tree.

  (i) No distance shrinks. (dominating)

Approximate metric using a tree.

Tree metric:
 X is nodes of tree with edge weights
 $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
 Tree weights are geometrically decreasing.

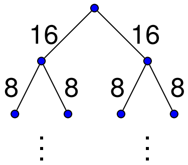


Map X into tree.
 (i) No distance shrinks. (dominating)
 (ii) Every distance stretches $\leq \alpha$

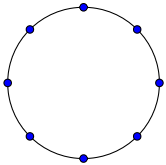
Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.



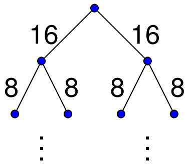
Map X into tree.
 (i) No distance shrinks. (dominating)
 (ii) Every distance stretches $\leq \alpha$



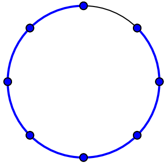
Map metric onto tree?

Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.



Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.



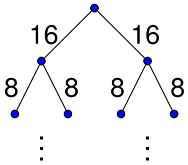
Map X into tree.
 (i) No distance shrinks. (dominating)
 (ii) Every distance stretches $\leq \alpha$

Map metric onto tree?

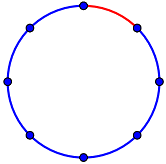
Approximate metric using a tree.

Tree metric:
X is nodes of tree with edge weights
$d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
Tree weights are geometrically decreasing.



Map X into tree.
(i) No distance shrinks. (dominating)
(ii) Every distance stretches $\leq \alpha$



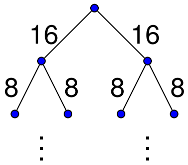
Distance 1 goes to $n-1$!

Map metric onto tree?

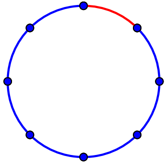
Approximate metric using a tree.

Tree metric:
X is nodes of tree with edge weights
$d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
Tree weights are geometrically decreasing.



Map X into tree.
(i) No distance shrinks. (dominating)
(ii) Every distance stretches $\leq \alpha$



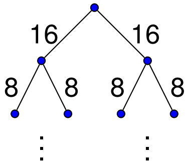
Map metric onto tree?

Distance 1 goes to $n-1$!
Bummer.

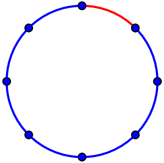
Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.

Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.



Probabilistic Tree embedding.
  Map X into tree.
  (i) No distance shrinks. (dominating)
  (ii) Every distance stretches $\leq \alpha$
    in expectation.



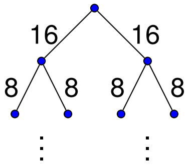
Distance 1 goes to $n-1$!
  Bummer.

Map metric onto tree?

Fix it up chappie!

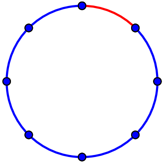
Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.



Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
  Map X into tree.
  (i) No distance shrinks. (dominating)
  (ii) Every distance stretches $\leq \alpha$
    in expectation.



Distance 1 goes to $n-1$!
  Bummer.

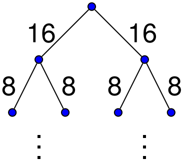
Map metric onto tree?

Fix it up chappie!

For cycle, remove a random edge

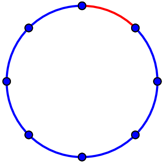
Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.



Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
  Map X into tree.
   (i) No distance shrinks. (dominating)
   (ii) Every distance stretches $\leq \alpha$
      in expectation.



Distance 1 goes to $n-1$!
  Bummer.

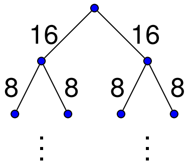
Map metric onto tree?

Fix it up chappie!

For cycle, remove a random edge get a tree.

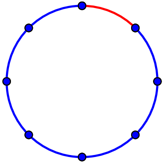
Approximate metric using a tree.

Tree metric:
X is nodes of tree with edge weights
$d_T(i,j)$ shortest path metric on tree.



Hierarchically well separated tree metric:
Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks. (dominating)
(ii) Every distance stretches $\leq \alpha$
in expectation.



Distance 1 goes to $n-1$!
Bummer.

Map metric onto tree?

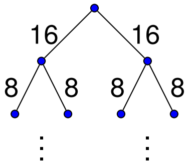
Fix it up chappie!

For cycle, remove a random edge get a tree.

Stretch of edge: $\frac{n-1}{n} \times 1$

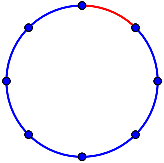
Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.



Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
  Map X into tree.
  (i) No distance shrinks. (dominating)
  (ii) Every distance stretches $\leq \alpha$
    in expectation.



Distance 1 goes to $n-1$!
  Bummer.

Map metric onto tree?

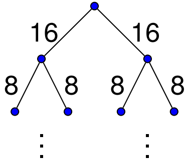
Fix it up chappie!

For cycle, remove a random edge get a tree.

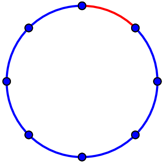
Stretch of edge: $\frac{n-1}{n} \times 1 + \frac{1}{n} \times (n-1)$

Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.



Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.



Probabilistic Tree embedding.
  Map X into tree.
  (i) No distance shrinks. (dominating)
  (ii) Every distance stretches $\leq \alpha$
    in expectation.

Distance 1 goes to $n-1$!
  Bummer.

Map metric onto tree?

  Fix it up chappie!

  For cycle, remove a random edge get a tree.

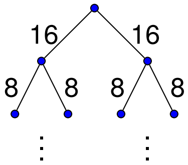
  Stretch of edge: $\frac{n-1}{n} \times 1 + \frac{1}{n} \times (n-1) \approx$

Approximate metric using a tree.

Tree metric:

X is nodes of tree with edge weights

$d_T(i,j)$ shortest path metric on tree.



Hierarchically well separated tree metric:

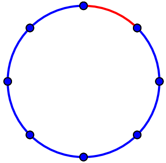
Tree weights are geometrically decreasing.

Probabilistic Tree embedding.

Map X into tree.

(i) No distance shrinks. (dominating)

(ii) Every distance stretches $\leq \alpha$ in expectation.



Distance 1 goes to $n-1$!

Bummer.

Map metric onto tree?

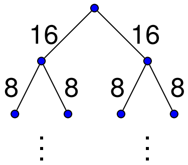
Fix it up chappie!

For cycle, remove a random edge get a tree.

Stretch of edge: $\frac{n-1}{n} \times 1 + \frac{1}{n} \times (n-1) \approx 2$

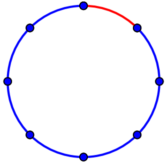
Approximate metric using a tree.

Tree metric:
  X is nodes of tree with edge weights
  $d_T(i,j)$ shortest path metric on tree.



Hierarchically well separated tree metric:
  Tree weights are geometrically decreasing.

Probabilistic Tree embedding.
  Map X into tree.
  (i) No distance shrinks. (dominating)
  (ii) Every distance stretches $\leq \alpha$
    in expectation.



Distance 1 goes to $n-1$!
  Bummer.

Map metric onto tree?

Fix it up chappie!

For cycle, remove a random edge get a tree.

Stretch of edge: $\frac{n-1}{n} \times 1 + \frac{1}{n} \times (n-1) \approx 2$

General metrics?

Probabilistic Tree embedding.

Probabilistic Tree embedding.

Map X into tree.

Probabilistic Tree embedding.

Probabilistic Tree embedding.

Map X into tree.

(i) No distance shrinks (dominating).

Probabilistic Tree embedding.

Probabilistic Tree embedding.

Map X into tree.

(i) No distance shrinks (dominating).

(ii) Every distance stretches $\leq \alpha$ in expecation.

Probabilistic Tree embedding.

Probabilistic Tree embedding.

Map X into tree.

(i) No distance shrinks (dominating).

(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.

Probabilistic Tree embedding.

Probabilistic Tree embedding.

Map X into tree.

(i) No distance shrinks (dominating).

(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).

Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

The Idea:

HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.

Recurse on each ball for $\Delta/2$.

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$.

Use randomness in

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$.

Use randomness in
selection of ball centers.

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$.

Use randomness in
selection of ball centers.
the $\approx$ diameter of the balls.

Probabilistic Tree embedding.

Probabilistic Tree embedding.
Map X into tree.
(i) No distance shrinks (dominating).
(ii) Every distance stretches $\leq \alpha$ in expecation.

Today: the tree will be Hierarchically well-separated (HST).
Elements of X are leaves of tree.

Later: **use** spanning tree for graphical metrics.

The Idea:
HST $\equiv$ recursive decomposition of metric space.

Decompose space by diameter $\approx \Delta$ balls.
Recurse on each ball for $\Delta/2$.

Use randomness in
selection of ball centers.
the $\approx$ diameter of the balls.

Algorithm

Algorithm: (X,d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

Algorithm

Algorithm: (X, d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i, j) \geq 1$

1. π – random permutation of X.

Algorithm

Algorithm: (X, d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

Algorithm

Algorithm: (X, d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

  def subtree(S,Δ):

Algorithm

Algorithm: (X,d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
 def subtree(S,Δ):
  T = []
```

Algorithm

Algorithm: (X, d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
```

Algorithm

Algorithm: (X,d), diam$(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
```

Algorithm

Algorithm: (X,d), $\text{diam}(X) \le D$, $|X| = n$, $d(i,j) \ge 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
 def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
```

Algorithm

Algorithm: (X,d), diam$(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ)
```

Algorithm

Algorithm: (X, d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
```

Algorithm

Algorithm: (X,d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
      T.append(B)
```

Algorithm

Algorithm: (X, d), $\text{diam}(X) \le D$, $|X| = n$, $d(i,j) \ge 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
      T.append(B)
  return map (λ x: subtree(x,Δ/2), T);
```

Algorithm

Algorithm: (X, d), diam$(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
 def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
      T.append(B)
  return map (λ x: subtree(x,Δ/2), T);
```

3. subtree(X, D)

Algorithm

Algorithm: (X,d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
      T.append(B)
  return map (λ x: subtree(x,Δ/2), T);
```

3. subtree(X,D)

Tree has internal node for each level of call.

Algorithm

Algorithm: (X,d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
      T.append(B)
  return map (λ x: subtree(x,Δ/2), T);
```

3. subtree(X, D)

Tree has internal node for each level of call. Tree edges have weight Δ to children.

Algorithm

Algorithm: (X,d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
      T.append(B)
  return map (λ x: subtree(x,Δ/2), T);
```

3. subtree(X,D)

Tree has internal node for each level of call. Tree edges have weight Δ to children.

Claim 1: $d_T(x,y) \geq d(x,y)$.

Algorithm

Algorithm: (X,d), $\text{diam}(X) \le D$, $|X| = n$, $d(i,j) \ge 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
      T.append(B)
  return map (λ x: subtree(x,Δ/2), T);
```

3. subtree(X,D)

Tree has internal node for each level of call. Tree edges have weight Δ to children.

Claim 1: $d_T(x,y) \ge d(x,y)$.

When $\Delta \le d(x,y)$, x and y must be in different balls, so cut at lvl $\Delta \ge d(x,y)/2$.

Algorithm

Algorithm: (X,d), $\text{diam}(X) \leq D$, $|X| = n$, $d(i,j) \geq 1$

1. π – random permutation of X.
2. Choose β in $[\frac{3}{8}, \frac{1}{2}]$.

```
def subtree(S,Δ):
  T = []
  if Δ < 1 return [S]
  foreach i in π:
    if i ∈ S
      B = ball(i, βΔ) ; S = S/B
      T.append(B)
  return map (λ x: subtree(x,Δ/2), T);
```

3. subtree(X,D)

Tree has internal node for each level of call. Tree edges have weight Δ to children.

Claim 1: $d_T(x,y) \geq d(x,y)$.

When $\Delta \leq d(x,y)$, x and y must be in different balls, so cut at lvl $\Delta \geq d(x,y)/2$.

$\rightarrow d_T(x,y) \geq \Delta + \Delta \geq d(x,y)$

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n)d(x,y)$.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n)d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

$Pr[\text{cut at level}\Delta]$?

Would like it to be $\frac{d(x,y)}{\Delta}$.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i} (4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n)d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i}(4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta = D/2^i} (4\Delta)\frac{d(x,y)}{\Delta} = 4 \log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?

smaller the edge the less likely to be on edge of ball.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n)d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i}(4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?

smaller the edge the less likely to be on edge of ball.

larger the delta, more room inside ball.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta = D/2^i} (4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?

smaller the edge the less likely to be on edge of ball.

larger the delta, more room inside ball.

random diameter jiggles edge of ball.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta = D/2^i} (4\Delta) \frac{d(x,y)}{\Delta} = 4 \log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?

smaller the edge the less likely to be on edge of ball.

larger the delta, more room inside ball.

random diameter jiggles edge of ball.

$\rightarrow Pr[x,y$ cut by ball$|x$ in ball$] \approx \frac{d(x,y)}{\beta\Delta}$

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i} (4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?
smaller the edge the less likely to be on edge of ball.
larger the delta, more room inside ball.
random diameter jiggles edge of ball.

$\rightarrow Pr[x, y$ cut by ball$|x$ in ball$] \approx \frac{d(x,y)}{\beta\Delta}$

The problem?

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n) d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i}(4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?
smaller the edge the less likely to be on edge of ball.
larger the delta, more room inside ball.
random diameter jiggles edge of ball.

$\rightarrow Pr[x,y$ cut by ball$|x$ in ball$] \approx \frac{d(x,y)}{\beta\Delta}$

The problem?
Could be cut be many different balls.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n)d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

Pr[cut at levelΔ]?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i}(4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?
smaller the edge the less likely to be on edge of ball.
larger the delta, more room inside ball.
random diameter jiggles edge of ball.

$\rightarrow Pr[x,y$ cut by ball$|x$ in ball$] \approx \frac{d(x,y)}{\beta\Delta}$

The problem?
Could be cut be many different balls.
For each probability is good, but could be hit by many.

Analysis: idea

Claim: $E[d_T(x,y)] = O(\log n)d(x,y)$.

Cut at level $\Delta \rightarrow d_T(x,y) \leq 4\Delta$. (Level of subtree call.)

$Pr[\text{cut at level}\Delta]$?

Would like it to be $\frac{d(x,y)}{\Delta}$.

$\rightarrow$ expected length is $\sum_{\Delta=D/2^i}(4\Delta)\frac{d(x,y)}{\Delta} = 4\log D \cdot d(x,y)$.

Why should it be $\frac{d(x,y)}{\Delta}$?
smaller the edge the less likely to be on edge of ball.
larger the delta, more room inside ball.
random diameter jiggles edge of ball.

$\rightarrow Pr[x,y \text{ cut by ball} | x \text{ in ball}] \approx \frac{d(x,y)}{\beta\Delta}$

The problem?
Could be cut be many different balls.
For each probability is good, but could be hit by many.
random permutation to deal with this

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$

(Only consider cut by x, factor 2 loss.)

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x, y) if..

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x, y) if..
$d(j, x) \leq \beta\Delta$

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x, y) if..
$d(j, x) \leq \beta\Delta$ and $\beta\Delta \leq d(j, y)$

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x, y) if..
$d(j, x) \leq \beta\Delta$ and $\beta\Delta \leq d(j, y) \leq d(j, x) + d(x, y)$

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x, y) if..
$d(j, x) \leq \beta\Delta$ and $\beta\Delta \leq d(j, y) \leq d(j, x) + d(x, y)$
$\rightarrow \beta\Delta \in [d[j, x], d(j, x) + d(x, y)]$.

Analysis: (x,y)

Would like $Pr[x,y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x,y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x,y) if $d(j,x) \in [\Delta/4, \Delta/2]$ (else (x,y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x,y) if..
$d(j,x) \leq \beta\Delta$ and $\beta\Delta \leq d(j,y) \leq d(j,x) + d(x,y)$
$\rightarrow \beta\Delta \in [d[j,x], d(j,x) + d(x,y)]$.
occurs with prob. $\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}$.

Analysis: (x,y)

Would like $Pr[x,y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x,y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x,y) if $d(j,x) \in [\Delta/4, \Delta/2]$ (else (x,y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x,y) if..
$d(j,x) \leq \beta\Delta$ and $\beta\Delta \leq d(j,y) \leq d(j,x) + d(x,y)$
$\rightarrow \beta\Delta \in [d[j,x], d(j,x) + d(x,y)]$.
occurs with prob. $\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}$.

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x, y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x, y) if..
$d(j, x) \leq \beta\Delta$ and $\beta\Delta \leq d(j, y) \leq d(j, x) + d(x, y)$
$\rightarrow \beta\Delta \in [d[j, x], d(j, x) + d(x, y)]$.
occurs with prob. $\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}$.

And j must be before any $i < j$ in π

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x,y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x, y) if..
$d(j,x) \leq \beta\Delta$ and $\beta\Delta \leq d(j,y) \leq d(j,x) + d(x,y)$
$\rightarrow \beta\Delta \in [d[j,x], d(j,x) + d(x,y)]$.
occurs with prob. $\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}$.

And j must be before any $i < j$ in $\pi \rightarrow$ prob is $\frac{1}{j}$

Analysis: (x,y)

Would like $Pr[x,y \text{ cut by ball}|x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x,y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x,y) if $d(j,x) \in [\Delta/4, \Delta/2]$ (else (x,y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x,y) if..
$d(j,x) \leq \beta\Delta$ and $\beta\Delta \leq d(j,y) \leq d(j,x) + d(x,y)$
$\rightarrow \beta\Delta \in [d[j,x], d(j,x) + d(x,y)]$.
occurs with prob. $\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}$.

And j must be before any $i < j$ in π $\rightarrow$ prob is $\frac{1}{j}$

$\rightarrow Pr[j \text{ cuts } (x,y)] \leq \left(\frac{1}{j}\right)\frac{8d(x,y)}{\Delta}$

Analysis: (x,y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x,y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x,y) if $d(j,x) \in [\Delta/4, \Delta/2]$ (else (x,y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x,y) if..
$d(j,x) \leq \beta\Delta$ and $\beta\Delta \leq d(j,y) \leq d(j,x) + d(x,y)$
$\rightarrow \beta\Delta \in [d[j,x], d(j,x) + d(x,y)]$.
occurs with prob. $\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}$.

And j must be before any $i < j$ in π $\rightarrow$ prob is $\frac{1}{j}$

$\rightarrow Pr[j \text{ cuts } (x,y)] \leq \left(\frac{1}{j}\right) \frac{8d(x,y)}{\Delta}$

$d_T(x,y)$ if cut level Δ is 4Δ.

Analysis: (x,y)

Would like $Pr[x,y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x,y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x,y) if $d(j,x) \in [\Delta/4, \Delta/2]$ (else (x,y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x,y) if..
$d(j,x) \leq \beta\Delta$ and $\beta\Delta \leq d(j,y) \leq d(j,x) + d(x,y)$
$\rightarrow \beta\Delta \in [d[j,x], d(j,x) + d(x,y)]$.
occurs with prob. $\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}$.

And j must be before any $i < j$ in π $\rightarrow$ prob is $\frac{1}{j}$

$\rightarrow Pr[j \text{ cuts } (x,y)] \leq \left(\frac{1}{j}\right) \frac{8d(x,y)}{\Delta}$

$d_T(x,y)$ if cut level Δ is 4Δ.

$\rightarrow E[d_T(x,y)] = \sum_{\Delta = \frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

Analysis: (x, y)

Would like $Pr[x, y \text{ cut by ball} | x \text{ in ball}] \leq \frac{8d(x,y)}{\Delta}$
(Only consider cut by x, factor 2 loss.)

At level Δ

At some point x is in some Δ level ball.
Renumber nodes in order of distance from x.

If $d(x,y) \geq \Delta/8$, $\frac{8d(x,y)}{\Delta} \geq 1$, so claim holds trivially.

j can only cut (x, y) if $d(j, x) \in [\Delta/4, \Delta/2]$ (else (x, y) entirely in ball),
Call this set X_Δ.

$j \in X_\Delta$ cuts (x, y) if..
$d(j, x) \leq \beta\Delta$ and $\beta\Delta \leq d(j, y) \leq d(j, x) + d(x, y)$
$\rightarrow \beta\Delta \in [d[j, x], d(j, x) + d(x, y)]$.
occurs with prob. $\frac{d(x,y)}{\Delta/8} = \frac{8d(x,y)}{\Delta}$.

And j must be before any $i < j$ in $\pi \rightarrow$ prob is $\frac{1}{j}$

$\rightarrow Pr[j \text{ cuts } (x, y)] \leq \left(\frac{1}{j}\right) \frac{8d(x,y)}{\Delta}$

$d_T(x, y)$ if cut level Δ is 4Δ.
$\rightarrow E[d_T(x, y)] = \sum_{\Delta = \frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x, y)$

The pipes are distinct!

$$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$$

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta = D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32 d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32 d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

"Listen Stash, the pipes are distinct!!"

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

“Listen Stash, the pipes are distinct!!”

Uh.. well X_Δ is distinct from $X_{\Delta/2}$.

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32 d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

“Listen Stash, the pipes are distinct!!”

Uh.. well X_Δ is distinct from $X_{\Delta/2}$.

$E(d_T(x,y)] = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32 d(x,y)$

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

"Listen Stash, the pipes are distinct!!"

Uh.. well X_Δ is distinct from $X_{\Delta/2}$.

$E(d_T(x,y)] = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

$\quad \leq \sum_j \left(\frac{1}{j}\right) 32d(x,y)$

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

"Listen Stash, the pipes are distinct!!"

Uh.. well X_Δ is distinct from $X_{\Delta/2}$.

$E(d_T(x,y)] = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

$\quad \leq \sum_j \left(\frac{1}{j}\right) 32d(x,y)$

$\leq (32 \ln n)(d(x,y))$.

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

"Listen Stash, the pipes are distinct!!"

Uh.. well X_Δ is distinct from $X_{\Delta/2}$.

$E(d_T(x,y)] = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$
$\leq \sum_j \left(\frac{1}{j}\right) 32d(x,y)$
$\leq (32 \ln n)(d(x,y))$.

Claim: $E[d_T(x,y)] = O(logn)d(x,y)$

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

"Listen Stash, the pipes are distinct!!"

Uh.. well X_Δ is distinct from $X_{\Delta/2}$.

$E(d_T(x,y)] = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$
$\leq \sum_j \left(\frac{1}{j}\right) 32d(x,y)$
$\leq (32 \ln n)(d(x,y))$.

Claim: $E[d_T(x,y)] = O(logn)d(x,y)$

Expected stretch is $O(\log n)$.

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

"Listen Stash, the pipes are distinct!!"

Uh.. well X_Δ is distinct from $X_{\Delta/2}$.

$E(d_T(x,y)] = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$
$\leq \sum_j \left(\frac{1}{j}\right) 32d(x,y)$
$\leq (32 \ln n)(d(x,y))$.

Claim: $E[d_T(x,y)] = O(logn)d(x,y)$

Expected stretch is $O(\log n)$.

We gave an algorithm that produces a distribution of trees.

The pipes are distinct!

$E(d_T(x,y)] = \sum_{\Delta=D/2^i} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$

Recall X_Δ has nodes with $d(x,j) \in [\Delta/4, \Delta/2]$

“Listen Stash, the pipes are distinct!!”

Uh.. well X_Δ is distinct from $X_{\Delta/2}$.

$E(d_T(x,y)] = \sum_{\Delta=\frac{D}{2^i}} \sum_{j \in X_\Delta} \left(\frac{1}{j}\right) 32d(x,y)$
$\leq \sum_j \left(\frac{1}{j}\right) 32d(x,y)$
$\leq (32 \ln n)(d(x,y))$.

Claim: $E[d_T(x,y)] = O(logn)d(x,y)$

Expected stretch is $O(\log n)$.

We gave an algorithm that produces a distribution of trees.

The expected stretch of any pair is $O(\log n)$.

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \to X$ that minimizes

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \to X$ that minimizes

$\sum_{e=(u,v)} c(e) d(l(u), l(v)) + \sum_v c(v, l(v))$

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \to X$ that minimizes

$\sum_{e=(u,v)} c(e) d(l(u), l(v)) + \sum_v c(v, l(v))$

Idea: find HST for metric (X, d).

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$\sum_{e=(u,v)} c(e)d(l(u), l(v)) + \sum_v c(v, l(v))$

Idea: find HST for metric (X, d).

Solve the problem on a hierarchically well separated tree metric.

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \to X$ that minimizes

$\sum_{e=(u,v)} c(e) d(l(u), l(v)) + \sum_v c(v, l(v))$

Idea: find HST for metric (X, d).

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \to X$ that minimizes

$\sum_{e=(u,v)} c(e)d(l(u), l(v)) + \sum_v c(v, l(v))$

Idea: find HST for metric (X, d).

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Hierarchically well separated tree,

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \to X$ that minimizes

$\sum_{e=(u,v)} c(e) d(l(u), l(v)) + \sum_v c(v, l(v))$

Idea: find HST for metric (X, d).

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Hierarchically well separated tree, “geometric”,

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$\sum_{e=(u,v)} c(e) d(l(u), l(v)) + \sum_v c(v, l(v))$

Idea: find HST for metric (X, d).

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Hierarchically well separated tree, “geometric”, constant factor.

Metric Labelling

Input: graph $G = (V, E)$ with edge weights, $w(\cdot)$, metric labels (X, d), and costs for mapping vertices to labels $c : V \times X$.

Find an labeling of vertices, $\ell : V \rightarrow X$ that minimizes

$\sum_{e=(u,v)} c(e) d(l(u), l(v)) + \sum_v c(v, l(v))$

Idea: find HST for metric (X, d).

Solve the problem on a hierarchically well separated tree metric.

Kleinberg-Tardos: constant factor on uniform metric.

Hierarchically well separated tree, “geometric”, constant factor.

$\rightarrow O(\log n)$ approximation.

And Now For Something...

Completely Different.

Example Problem: clustering.

- Points: documents, dna, preferences.
- Graphs: applications to VLSI, parallel processing, image segmentation.

Image example.

Image Segmentation

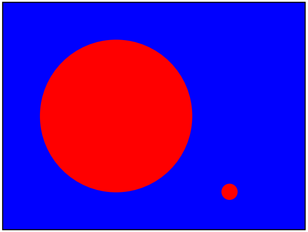
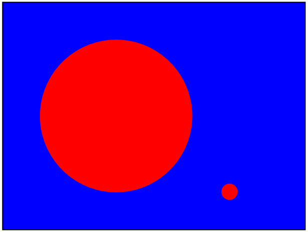


Image Segmentation



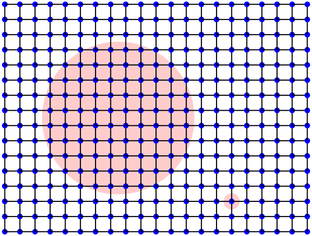
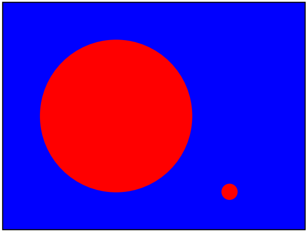


Image Segmentation



Which region?

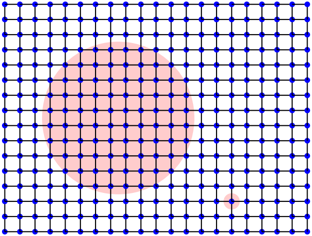
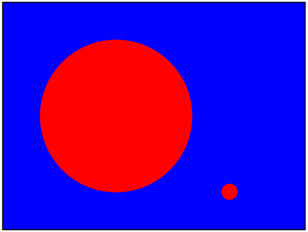
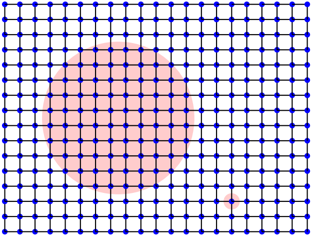


Image Segmentation

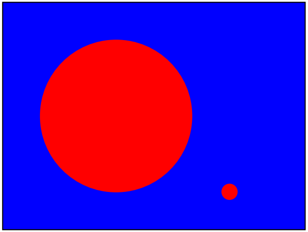


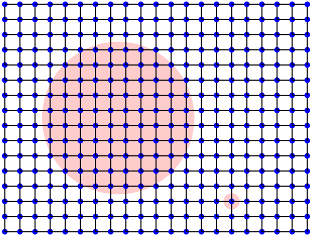


Which region?   Normalized Cut: Find S, which minimizes

$$\frac{w(S,\overline{S})}{w(S) \times w(\overline{S})}.$$

Image Segmentation





Which region? Normalized Cut: Find S, which minimizes

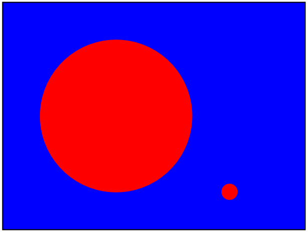
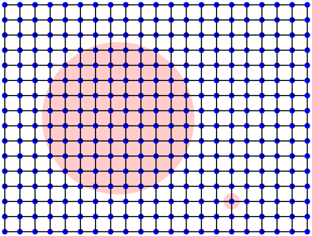
$$\frac{w(S,\overline{S})}{w(S)\times w(\overline{S})}.$$

Ratio Cut: minimize

$$\frac{w(S,\overline{S})}{w(S)},$$

$w(S)$ no more than half the weight. (Minimize cost per unit weight that is removed.)

Image Segmentation

Which region? Normalized Cut: Find S, which minimizes

$$\frac{w(S,\overline{S})}{w(S) \times w(\overline{S})}.$$

Ratio Cut: minimize

$$\frac{w(S,\overline{S})}{w(S)},$$

$w(S)$ no more than half the weight. (Minimize cost per unit weight that is removed.)

Either is generally useful!

Edge Expansion/Conductance.

Graph $G = (V, E)$,

Edge Expansion/Conductance.

Graph $G = (V, E)$,

Assume regular graph of degree d.

Edge Expansion/Conductance.

Graph $G = (V, E)$,

Assume regular graph of degree d.

Edge Expansion.

Edge Expansion/Conductance.

Graph $G = (V, E)$,

Assume regular graph of degree d.

Edge Expansion.

$h(S) = \frac{|E(S, V-S)|}{d \min |S|, |V-S|}$, $h(G) = \min_S h(S)$

Edge Expansion/Conductance.

Graph $G = (V, E)$,

Assume regular graph of degree d.

Edge Expansion.

$h(S) = \frac{|E(S,V-S)|}{d \min |S|,|V-S|}$, $h(G) = \min_S h(S)$

Conductance.

Edge Expansion/Conductance.

Graph $G = (V, E)$,

Assume regular graph of degree d.

Edge Expansion.

$h(S) = \frac{|E(S,V-S)|}{d \min |S|,|V-S|}$, $h(G) = \min_S h(S)$

Conductance.

$\phi(S) = \frac{n|E(S,V-S)|}{d|S||V-S|}$, $\phi(G) = \min_S \phi(S)$

Edge Expansion/Conductance.

Graph $G = (V, E)$,

Assume regular graph of degree d.

Edge Expansion.

$h(S) = \frac{|E(S,V-S)|}{d \min |S|,|V-S|}$, $h(G) = \min_S h(S)$

Conductance.

$\phi(S) = \frac{n|E(S,V-S)|}{d|S||V-S|}$, $\phi(G) = \min_S \phi(S)$

Note $n \geq \max(|S|, |V| - |S|) \geq n/2$

Edge Expansion/Conductance.

Graph $G = (V, E)$,

Assume regular graph of degree d.

Edge Expansion.

$h(S) = \frac{|E(S,V-S)|}{d \min |S|,|V-S|}$, $h(G) = \min_S h(S)$

Conductance.

$\phi(S) = \frac{n|E(S,V-S)|}{d|S||V-S|}$, $\phi(G) = \min_S \phi(S)$

Note $n \geq \max(|S|, |V| - |S|) \geq n/2$

$\rightarrow h(G) \leq \phi(G) \leq 2h(S)$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T M v' = v^T (\lambda' v')$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T M v' = v^T (\lambda' v') = \lambda' v^T v'$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T Mv' = v^T(\lambda' v') = \lambda' v^T v'$

$v^T Mv' = \lambda v^T v'$

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T Mv' = v^T(\lambda' v') = \lambda' v^T v'$

$v^T M v' = \lambda v^T v' = \lambda v^T v.$ □
Distinct eigenvalues

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T Mv' = v^T(\lambda' v') = \lambda' v^T v'$

$v^T Mv' = \lambda v^T v' = \lambda v^T v$. $\square$
Distinct eigenvalues $\rightarrow$ orthonormal basis.

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T M v' = v^T (\lambda' v') = \lambda' v^T v'$

$v^T M v' = \lambda v^T v' = \lambda v^T v.$ □

Distinct eigenvalues $\rightarrow$ orthonormal basis.

In basis: matrix is diagonal..

Spectra of the graph.

$M = A/d$ adjacency matrix, A

Eigenvector: $v - Mv = \lambda v$

Real, symmetric.

Claim: Any two eigenvectors with different eigenvalues are orthogonal.

Proof: Eigenvectors: v, v' with eigenvalues λ, λ'.
$v^T M v' = v^T (\lambda' v') = \lambda' v^T v'$

$v^T M v' = \lambda v^T v' = \lambda v^T v.$ □
Distinct eigenvalues $\rightarrow$ orthonormal basis.

In basis: matrix is diagonal..

$$M = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Action of M.

v - assigns weights to vertices.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value?

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i$

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i$

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.

Action of M.

v - assigns weights to vertices.

Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.

Action of M.

v - assigns weights to vertices.

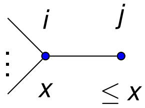
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



Action of M.

v - assigns weights to vertices.

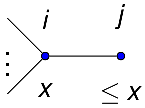
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j)$$

Action of M.

v - assigns weights to vertices.

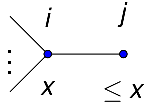
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Action of M.

v - assigns weights to vertices.

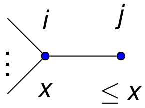
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$.

Action of M.

v - assigns weights to vertices.

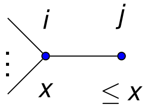
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1.$

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x.$

Therefore $\lambda_2 < 1$. □

Action of M.

v - assigns weights to vertices.

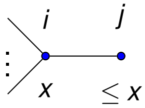
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.

Action of M.

v - assigns weights to vertices.

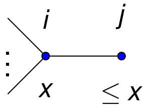
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.

Action of M.

v - assigns weights to vertices.

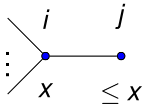
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$(Mv)_i \le \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i)$

Action of M.

v - assigns weights to vertices.

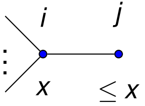
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i) \implies$ eigenvector with $\lambda = 1$.

Action of M.

v - assigns weights to vertices.

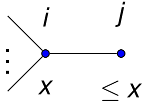
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j),\ v_i = x,\ x_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i) \implies$ eigenvector with $\lambda = 1$.
Choose δ to make $\sum_i x_i = 0$,

Action of M.

v - assigns weights to vertices.

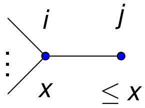
Mv replaces v_i with $\frac{1}{d}\sum_{e=(i,j)} v_j$.

Eigenvector with highest value? $v = \mathbf{1}$. $\lambda_1 = 1$.

$\rightarrow v_i = (M\mathbf{1})_i = \frac{1}{d}\sum_{e\in(i,j)} 1 = 1$.

Claim: For a connected graph $\lambda_2 < 1$.

Proof: Second Eigenvector: $v \perp \mathbf{1}$. Max value x.
Connected $\rightarrow$ path from x valued node to lower value.
$\rightarrow \exists\ e = (i,j)$, $v_i = x$, $x_j < x$.



$(Mv)_i \leq \frac{1}{d}(x + x \cdots + v_j) < x$.

Therefore $\lambda_2 < 1$. □

Claim: Connected if $\lambda_2 < 1$.
Proof: Assign $+1$ to vertices in one component, $-\delta$ to rest.
$x_i = (Mx_i) \implies$ eigenvector with $\lambda = 1$.
Choose δ to make $\sum_i x_i = 0$, i.e., $x \perp \mathbf{1}$. □

Rayleigh Quotient

$$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$$

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

xMx

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2$

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T Mx}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh quotient.

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. $\square$

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

$x \perp 1$

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

$x \perp 1 \leftrightarrow \sum_i x_i = 0$.

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

$x \perp 1 \leftrightarrow \sum_i x_i = 0$.

Example: $0/1$ Indicator vector for balanced cut, S is one such vector.

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T Mx}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T Mx}{x^T x}$.

$x \perp 1 \leftrightarrow \sum_i x_i = 0$.

Example: $0/1$ Indicator vector for balanced cut, S is one such vector.

Rayleigh quotient is $\frac{|E(S,S)|}{|S|}$

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T Mx}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T Mx}{x^T x}$.

$x \perp 1 \leftrightarrow \sum_i x_i = 0$.

Example: $0/1$ Indicator vector for balanced cut, S is one such vector.

Rayleigh quotient is $\frac{|E(S,S)|}{|S|} = h(S)$.

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

$x \perp 1 \leftrightarrow \sum_i x_i = 0$.

Example: $0/1$ Indicator vector for balanced cut, S is one such vector.

Rayleigh quotient is $\frac{|E(S,S)|}{|S|} = h(S)$.

Rayleigh quotient is less than $h(S)$ for any balanced cut S.

Rayleigh Quotient

$\lambda_1 = \max_x \frac{x^T M x}{x^T x}$

In basis, M is diagonal.

Represent x in basis, i.e., $x_i = x \cdot v_i$.

$xMx = \sum_i \lambda_i x_i^2 \leq \lambda_1 \sum_i x_i^2 \lambda = \lambda x^T x$

Tight when x is first eigenvector. □

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

$x \perp 1 \leftrightarrow \sum_i x_i = 0$.

Example: $0/1$ Indicator vector for balanced cut, S is one such vector.

Rayleigh quotient is $\frac{|E(S,S)|}{|S|} = h(S)$.

Rayleigh quotient is less than $h(S)$ for any balanced cut S.

Find balanced cut from vector that acheives Rayleigh quotient?

Cheeger's inequality.

Rayleigh quotient.

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2}$

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2}$

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G)$

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)}$

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Connected $\lambda_2 < \lambda_1$.

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Connected $\lambda_2 < \lambda_1$.

$h(G)$ large

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Connected $\lambda_2 < \lambda_1$.

$h(G)$ large $\rightarrow$ well connected

Cheeger's inequality.

Rayleigh quotient.
$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \le |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \le h(G) \le \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Connected $\lambda_2 < \lambda_1$.
$h(G)$ large $\rightarrow$ well connected $\rightarrow$ $\lambda_1 - \lambda_2$ big.

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Connected $\lambda_2 < \lambda_1$.

$h(G)$ large $\rightarrow$ well connected $\rightarrow$ $\lambda_1 - \lambda_2$ big.

Disconnected

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Connected $\lambda_2 < \lambda_1$.

$h(G)$ large $\rightarrow$ well connected $\rightarrow \lambda_1 - \lambda_2$ big.

Disconnected $\lambda_2 = \lambda_1$.

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Connected $\lambda_2 < \lambda_1$.

$h(G)$ large $\rightarrow$ well connected $\rightarrow$ $\lambda_1 - \lambda_2$ big.

Disconnected $\lambda_2 = \lambda_1$.

$h(G)$ small

Cheeger's inequality.

Rayleigh quotient.

$\lambda_2 = \max_{x \perp \mathbf{1}} \frac{x^T M x}{x^T x}$.

Eigenvalue gap: $\mu = \lambda_1 - \lambda_2$.

Recall: $h(G) = \min_{S, |S| \leq |V|/2} \frac{|E(S, V-S)|}{|S|}$

$\frac{\mu}{2} = \frac{1-\lambda_2}{2} \leq h(G) \leq \sqrt{2(1-\lambda_2)} = \sqrt{2\mu}$

Hmmm..

Connected $\lambda_2 < \lambda_1$.
$h(G)$ large $\rightarrow$ well connected $\rightarrow \lambda_1 - \lambda_2$ big.
Disconnected $\lambda_2 = \lambda_1$.
$h(G)$ small $\rightarrow \lambda_1 - \lambda_2$ small.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|)$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Same side endpoints: like $v^T v$.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S: v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Same side endpoints: like $v^T v$.

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S: v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$

$v^T M v = v^T v - (2|E(S,S)||S|(|V| - |S|)$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$

$v^T M v = v^T v - (2|E(S,S)||S|(|V| - |S|)$

$\frac{v^T M v}{v^T v} = 1 - \frac{2|E(S,\overline{S})|}{|S|}$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S : v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$

$v^T M v = v^T v - (2|E(S,S)||S|(|V| - |S|)$

$\frac{v^T M v}{v^T v} = 1 - \frac{2|E(S,\overline{S})|}{|S|}$

$\lambda_2 \geq 1 - 2h(S)$

Easy side of Cheeger.

Small cut $\rightarrow$ small eigenvalue gap.

$\frac{\mu}{2} \leq h(G)$

Cut S. $i \in S: v_i = |V| - |S|$, $i \in \overline{S} v_i = -|S|$.

$\sum_i v_i = |S|(|V| - |S|) - |S|(|V| - |S|) = 0$

$\rightarrow v \perp \mathbf{1}$.

$v^T v = |S|(|V| - |S|)^2 + |S|^2(|V| - |S|) = |S|(|V| - |S|)(|V|)$.

$v^T M v = \frac{1}{d} \sum_{e=(i,j)} x_i x_j$.

Same side endpoints: like $v^T v$.

Different side endpoints: $-|S|(|V| - |S|)$

$v^T M v = v^T v - (2|E(S,S)||S|(|V| - |S|)$

$\frac{v^T M v}{v^T v} = 1 - \frac{2|E(S,\overline{S})|}{|S|}$

$\lambda_2 \geq 1 - 2h(S) \rightarrow h(G) \geq \frac{1-\lambda_2}{2}$

See you ...

Thursday.