Approximating the Best–Fit Tree Under L_p Norms

Boulos Harb, Sampath Kannan and Andrew McGregor, UPenn

The Problem(s)

- Input: Distance Matrix *D*[*i,j*] on *n* items
- Output: Tree Metric *T*[*i,j*]
- Goal: Minimize the L_p cost-of-fit

$$L_p(D,T) = \left(\sum_{i,j} |D[i,j] - T[i,j]|^p \right)^{1/p}$$

The Problem(s)

- Input: Distance Matrix $D[i,j]$ on n items
- Output: Ultrametric $T[i,j]$
- Goal: Minimize the L_p cost-of-fit

$$L_p(D,T) = \left(\sum_{i,j} |D[i,j] - T[i,j]|^p \right)^{1/p}$$

The Problem(s)

- Input: Distance Matrix $D[i,j]$ on n items
- Output: Ultrametric $T[i,j]$
- Goal: Minimize the L_{rel} cost-of-fit

$$L_{\mathrm{rel}}(D,T) = \sum_{i,j} \max\left\{\frac{D[i,j]}{T[i,j]}, \frac{T[i,j]}{D[i,j]}\right\}$$

Tree Metric & Ultrametrics

- Tree Metric: Distances between the leaves of a weighted tree.

$$\forall w, x, y, z \in [n] \quad T[w,x] + T[y,z] \leq \max\{T[w,y] + T[x,z], T[w,z] + T[x,y]\}$$

- Ultrametric: Distance between the leaves of a rooted weighted tree in which all leaves are equidistance from root.

$$\forall x, y, z \in [n] \quad T[x,y] \leq \max\{T[x,z], T[z,y]\}$$

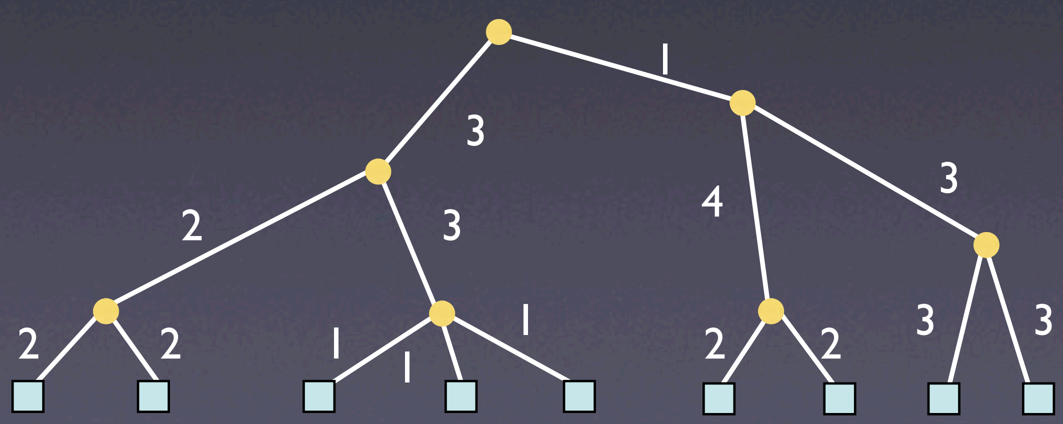
Tree Metric & Ultrametrics

- Tree Metric: Distances between the leaves of a weighted tree.

$$\forall w, x, y, z \in [n] \quad T[w,x] + T[y,z] \leq \max\{T[w,y] + T[x,z], T[w,z] + T[x,y]\}$$

- Ultrametric: Distance between the leaves of a rooted weighted tree in which all leaves are equidistance from root.

$$\forall x, y, z \in [n] \quad T[x,y] \leq \max\{T[x,z], T[z,y]\}$$

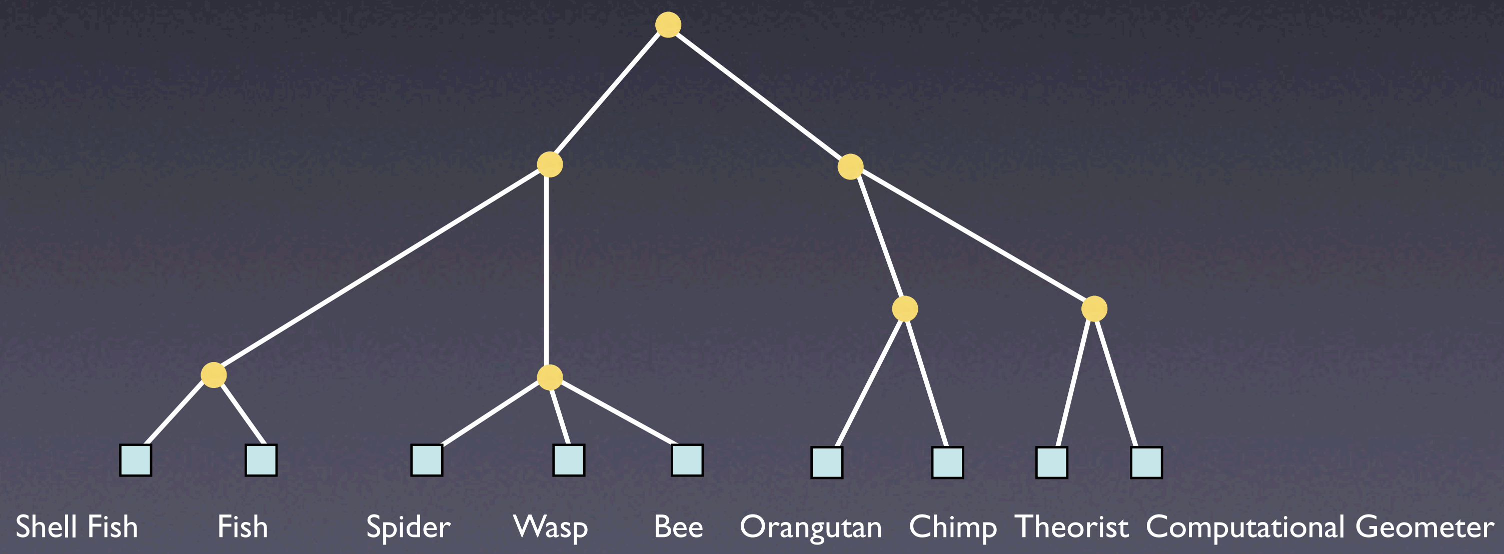


Biological Motivation

- View ultrametric as an evolutionary tree
- $D[i,j]$ is estimate of time since species i and j diverged
- Goal: Reconcile contradictory estimates

Biological Motivation

- View ultrametric as an evolutionary tree
- *D*[*i,j*] is estimate of time since species *i* and *j* diverged
- Goal: Reconcile contradictory estimates



Previous Work

Previous Work

- Farach, Kannan & Warnow '95:

  Exact construction of best-fit ultrametric under L_∞

Previous Work

- Farach, Kannan & Warnow '95:

  Exact construction of best-fit ultrametric under L_∞

- Agarwala, Bafna, Farach, Paterson & Thorup '99:

  3 approximation of best-fit tree under L_∞

Previous Work

- Farach, Kannan & Warnow '95:

  Exact construction of best-fit ultrametric under L_∞

- Agarwala, Bafna, Farach, Paterson & Thorup '99:

  3 approximation of best-fit tree under L_∞

- Ma, Wang & Zhang '99:

  $n^{1/p}$ approximation of best-fit non-contracting ultrametric under L_p

Previous Work

- Farach, Kannan & Warnow '95:

  Exact construction of best-fit ultrametric under L_∞

- Agarwala, Bafna, Farach, Paterson & Thorup '99:

  3 approximation of best-fit tree under L_∞

- Ma, Wang & Zhang '99:

  $n^{1/p}$ approximation of best-fit non-contracting ultrametric under L_p

- Dhamdhere '04:

  $O(\log^{1/p} n)$ approximation of best-fit line metric under L_p

Our Results

- Algorithm #1:

  L_p: $O(k \log n)^{1/p}$ approximation to best-fit tree where k is the number of distinct distances in D

  L_{rel}: $O(\log^2 n)$ approximation to best-fit ultrametric

- Algorithm #2:

  L_p: $n^{1/p}$ approximation to best-fit tree

Algorithm #1

Restricting Splitting Distances

Restricting Splitting Distances

- Original distances are $d_1 < d_2 < \ldots < d_k$

Restricting Splitting Distances

- Original distances are $d_1 < d_2 < \ldots < d_k$
- Lemma:

Restricting Splitting Distances

- Original distances are $d_1 < d_2 < \ldots < d_k$
- Lemma:

  a) There exists a best-fit (under L_1) ultrametric whose distances are a subset of $\{d_1, d_2, \ldots, d_k\}$

Restricting Splitting Distances

- Original distances are $d_1 < d_2 < \ldots < d_k$
- Lemma:

  a) There exists a best-fit (under L_1) ultrametric whose distances are a subset of $\{d_1, d_2, \ldots, d_k\}$

  b) There exists an ultrametric whose distances are a subset of $\{d_1, d_2, \ldots, d_k\}$ whose cost-of-fit is at most twice optimal (under L_p).

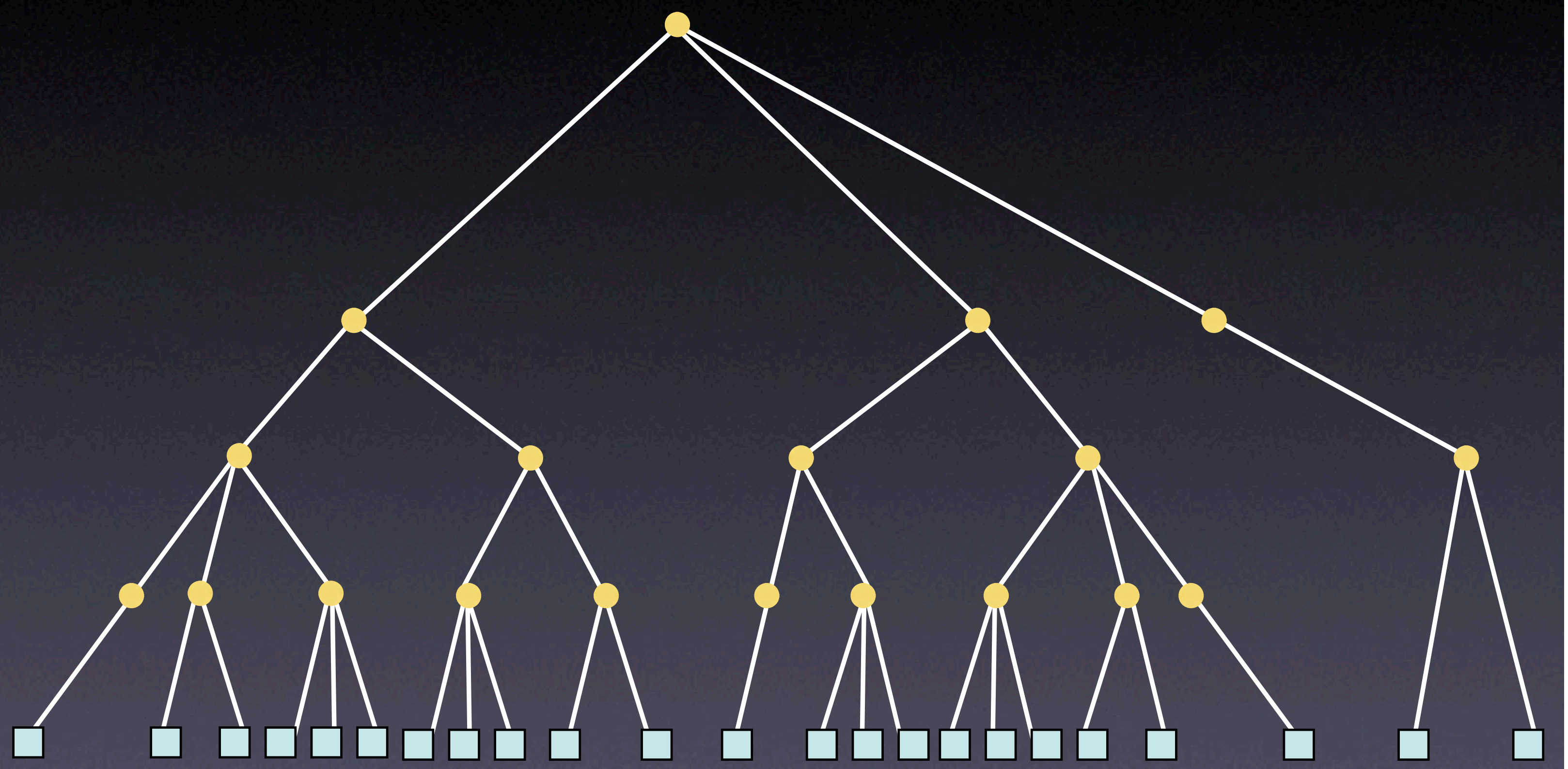
Restricting Splitting Distances

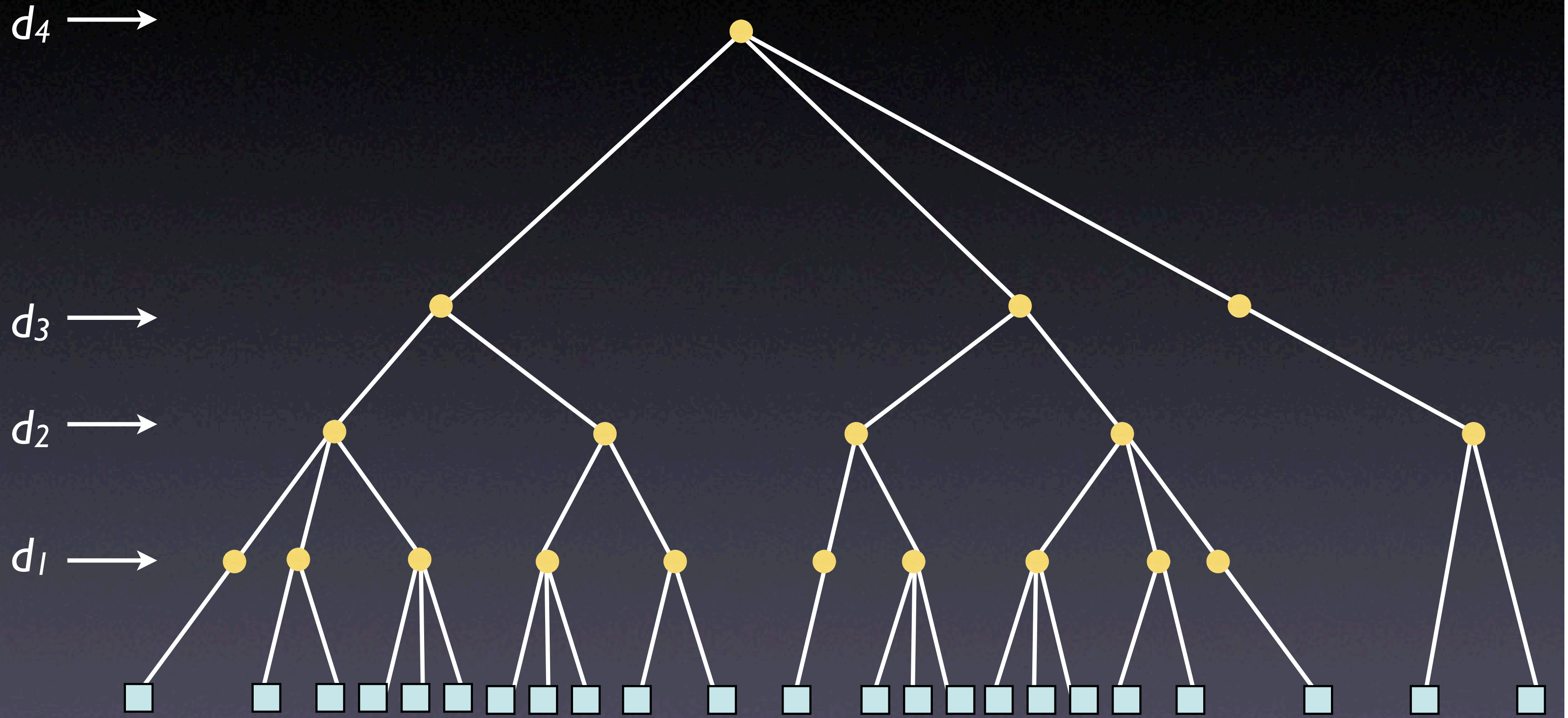
- Original distances are $d_1 < d_2 < \ldots < d_k$
- Lemma:

  a) There exists a best-fit (under L_1) ultrametric whose distances are a subset of $\{d_1, d_2, \ldots, d_k\}$

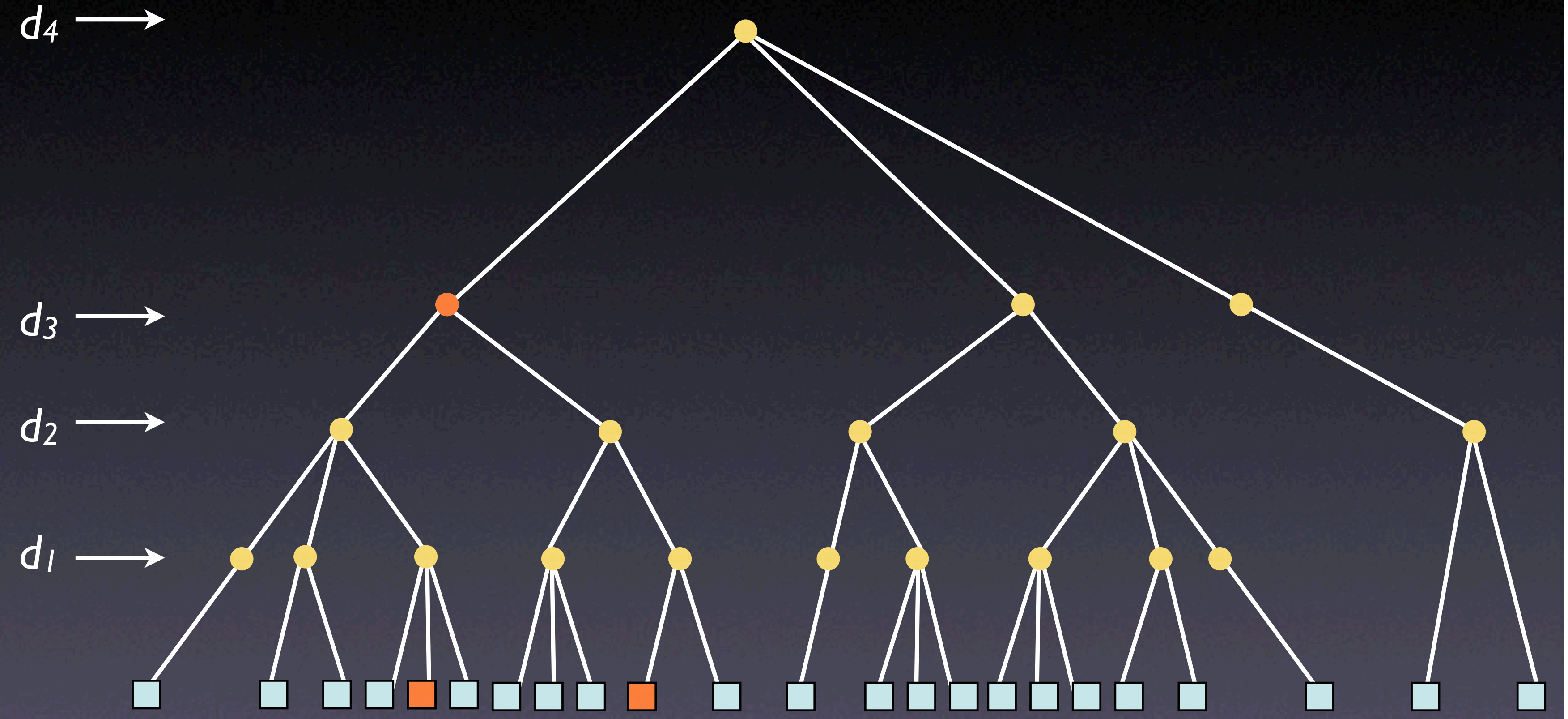
  b) There exists an ultrametric whose distances are a subset of $\{d_1, d_2, \ldots, d_k\}$ whose cost-of-fit is at most twice optimal (under L_p).

  c) There exists an ultrametric with O(log n) distances whose cost-of-fit is at most twice optimal (under L_{rel}). [Assuming d_k/d_1 is polynomial in n.]

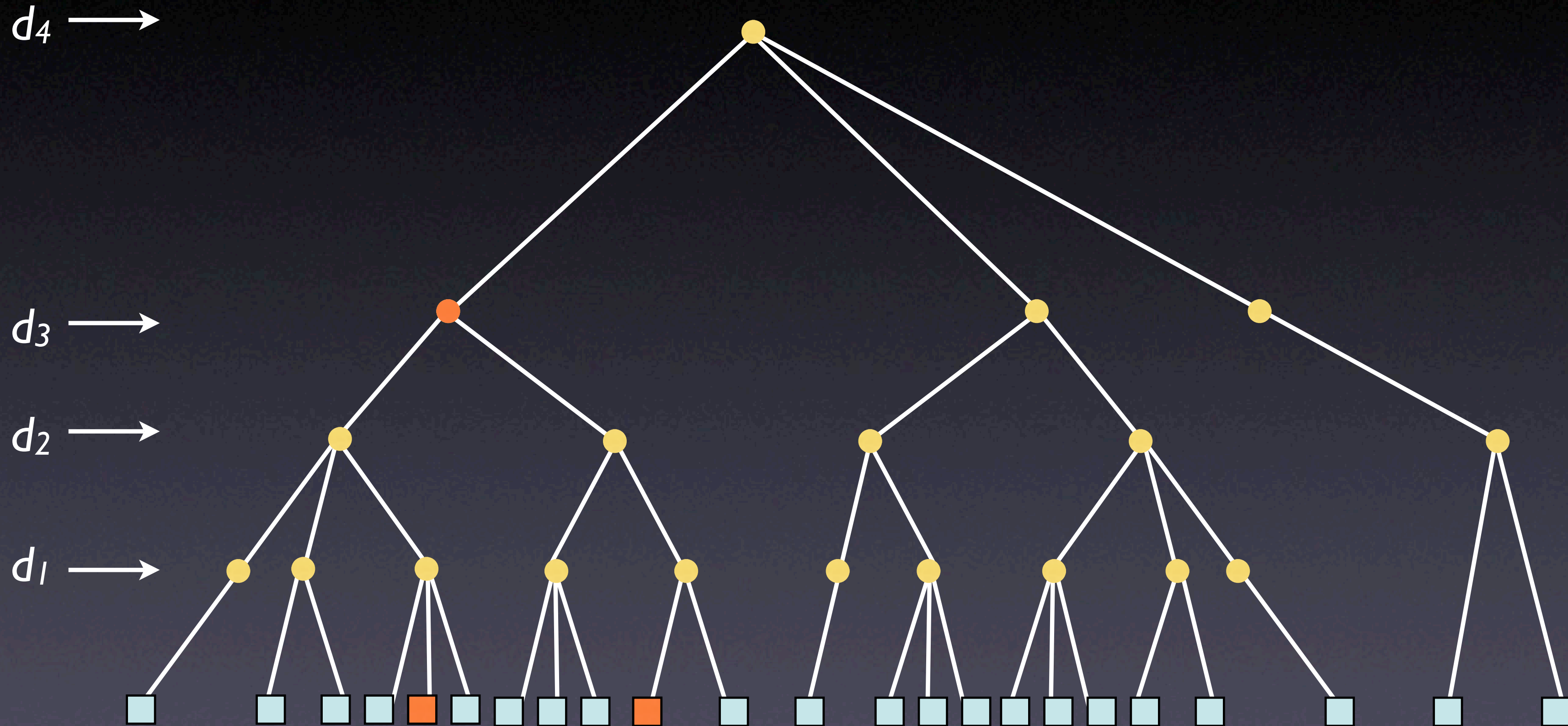




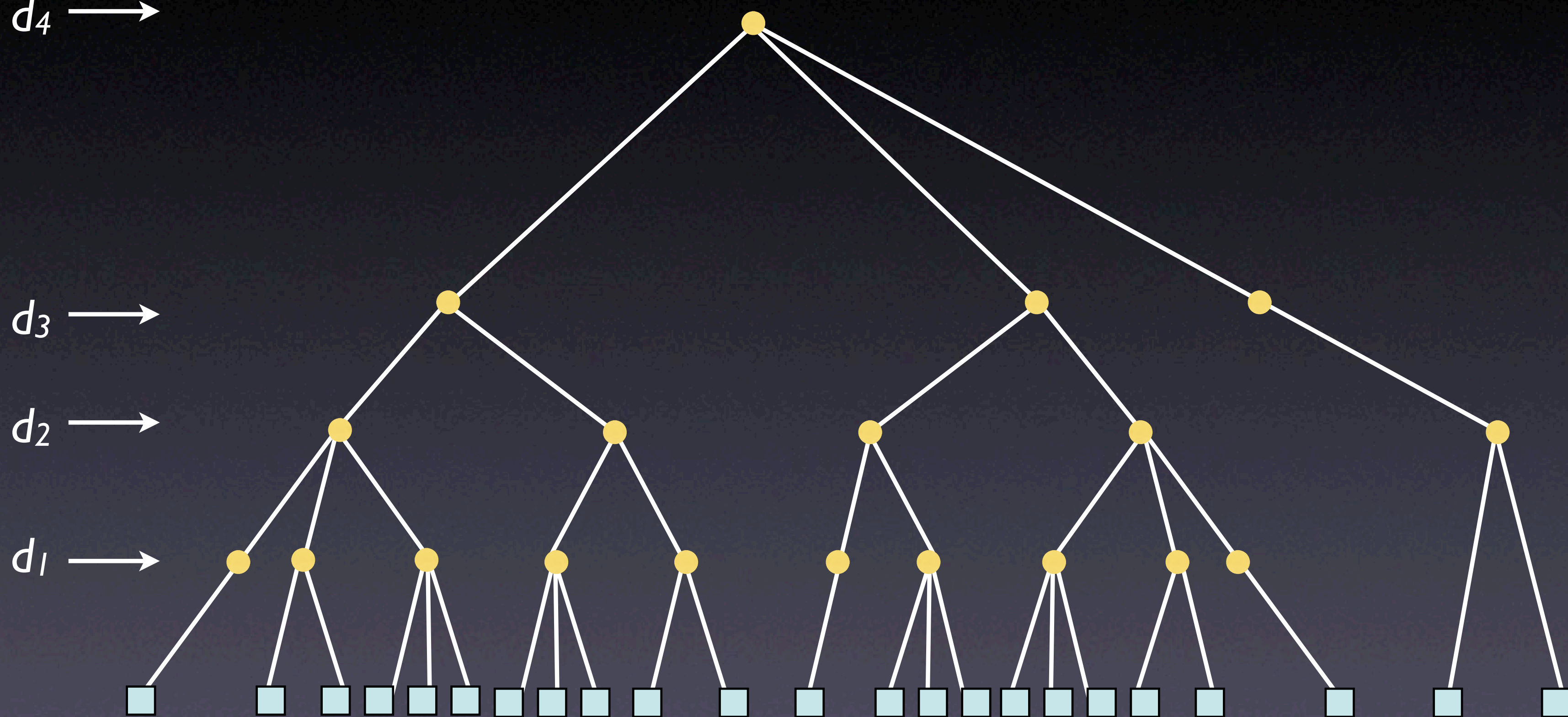
d_4
d_3
d_2
d_1



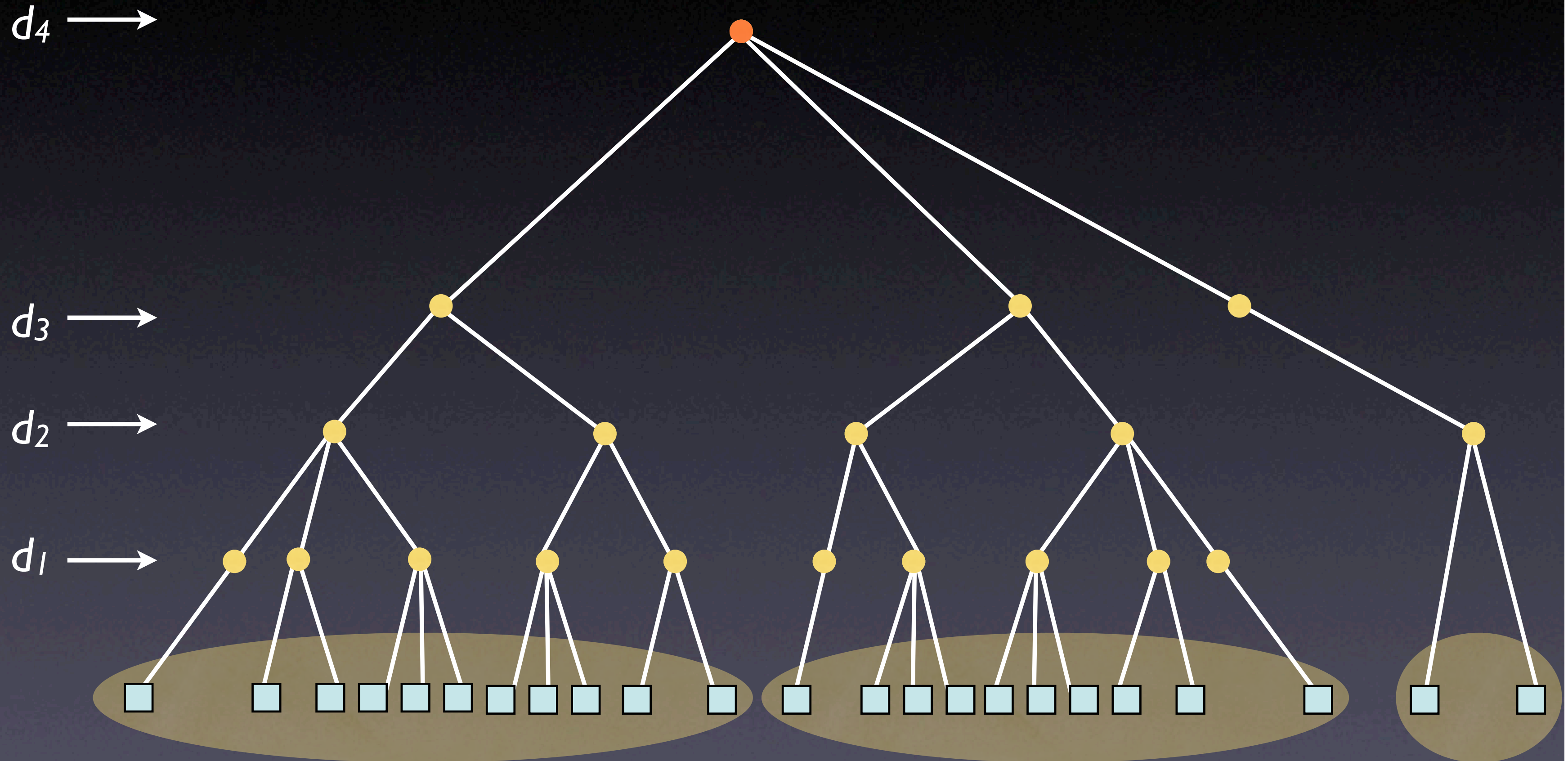
d_4
d_3
d_2
d_1



"Splitting Distance" of internal node v =
Distance between leaves of subtree rooted a v



"Splitting Distance" of internal node v =
Distance between leaves of subtree rooted a v



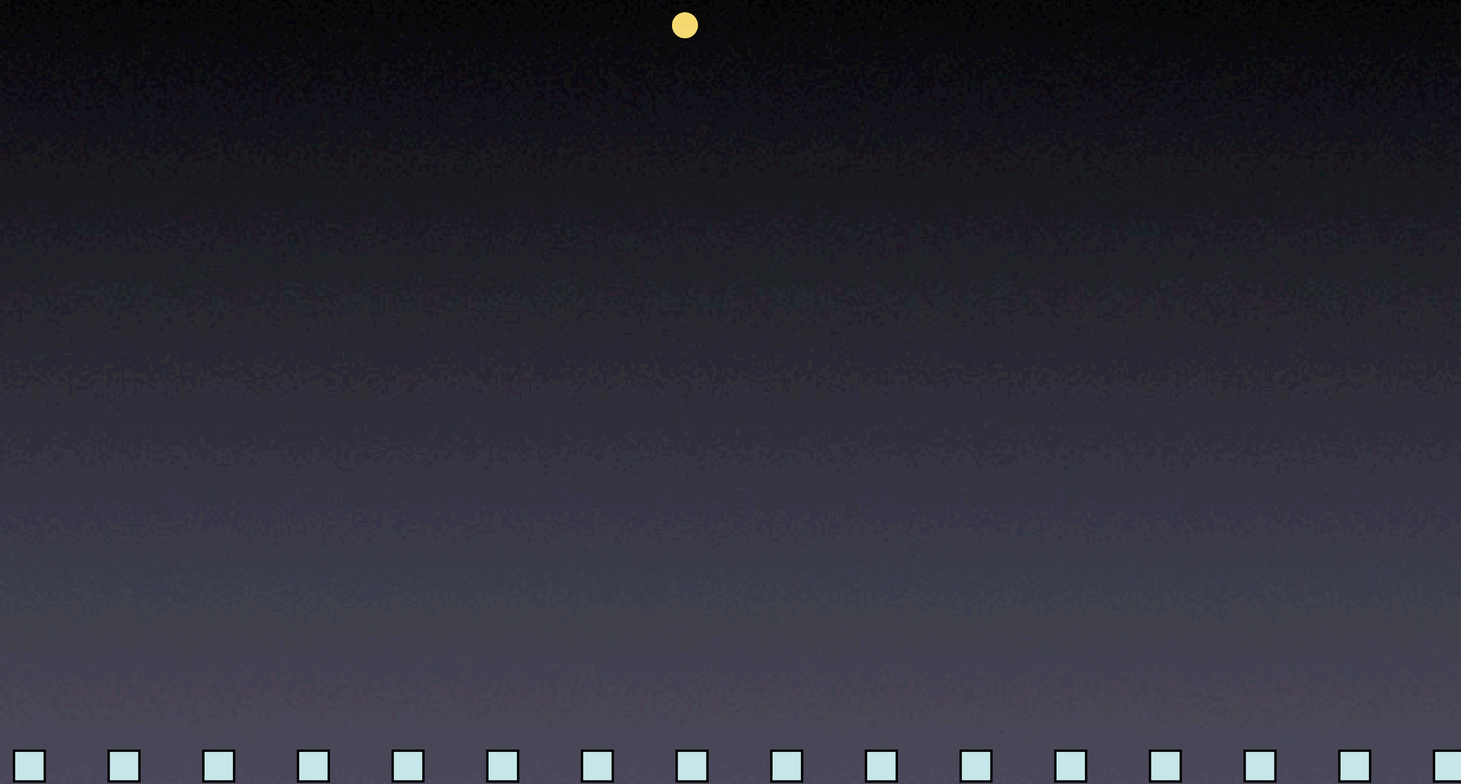
"Splitting Distance" of internal node v =
Distance between leaves of subtree rooted a v

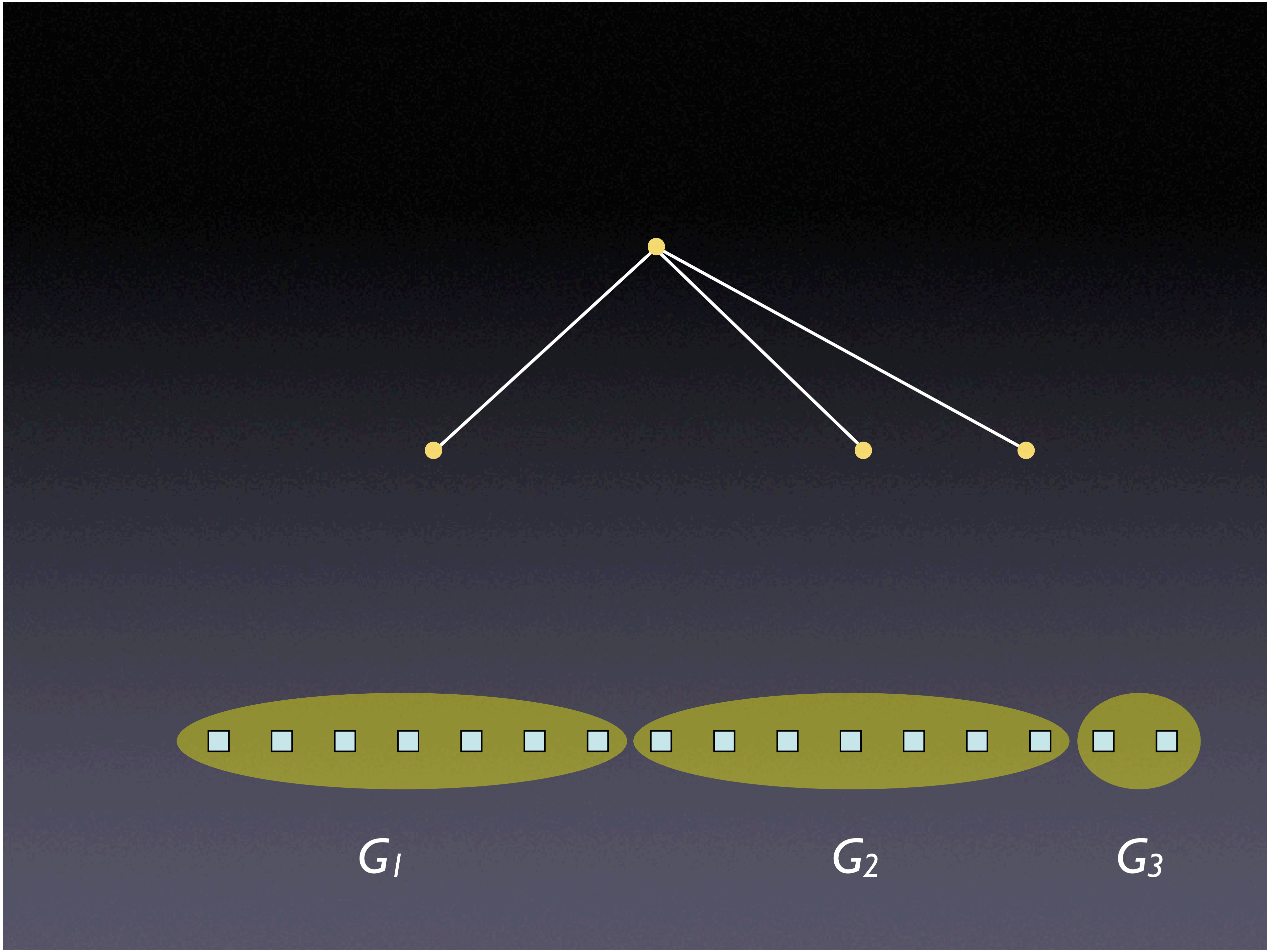
Algorithm Outline

- Construct top partition $G \rightarrow G_1, G_2, G_3, \ldots$

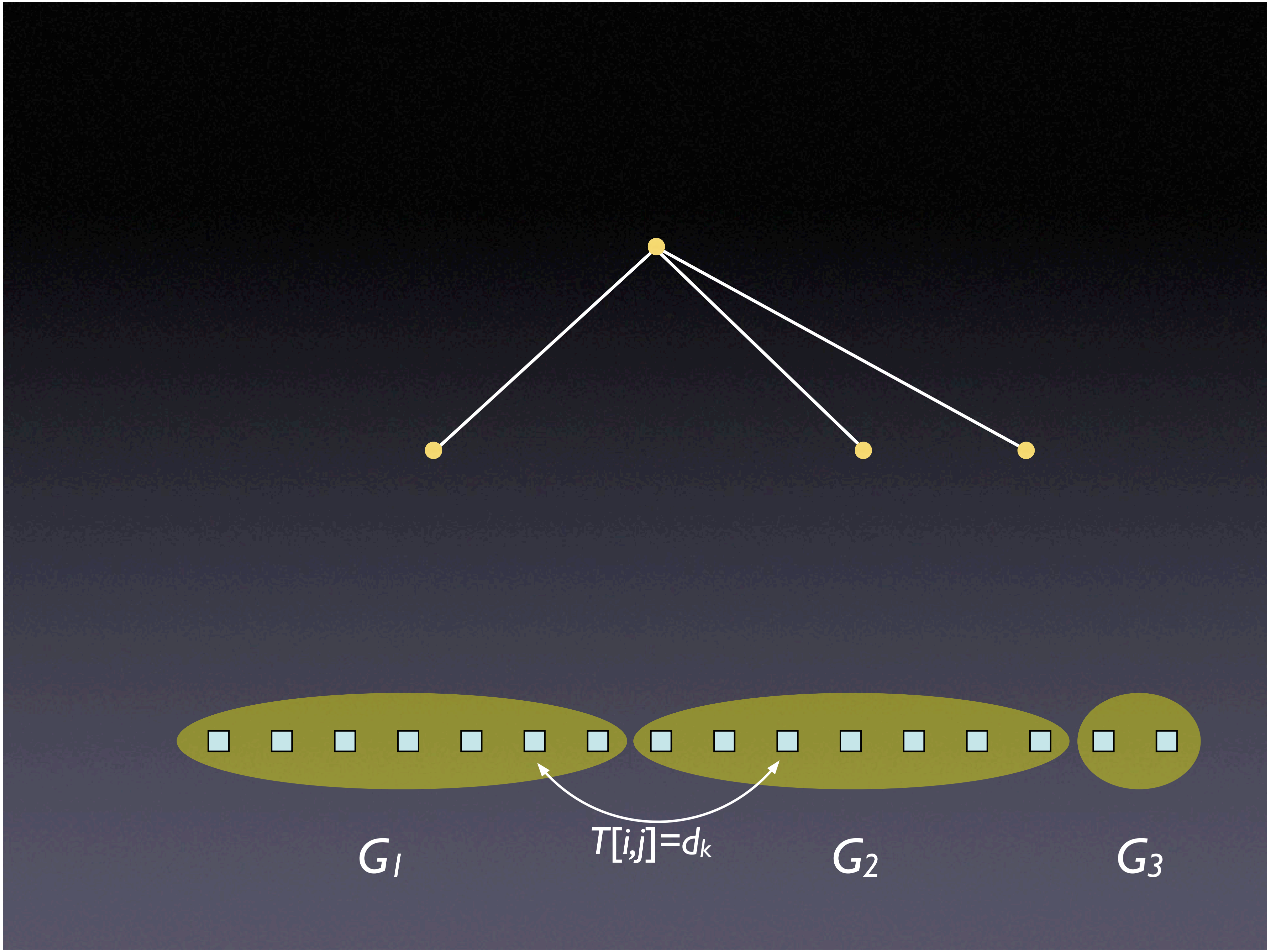
  Set length of inter-cluster edges to d_k

  All other lengths will be set to $\leq d_{k-1}$
- Construct trees for $G_1, G_2, G_3, \ldots$

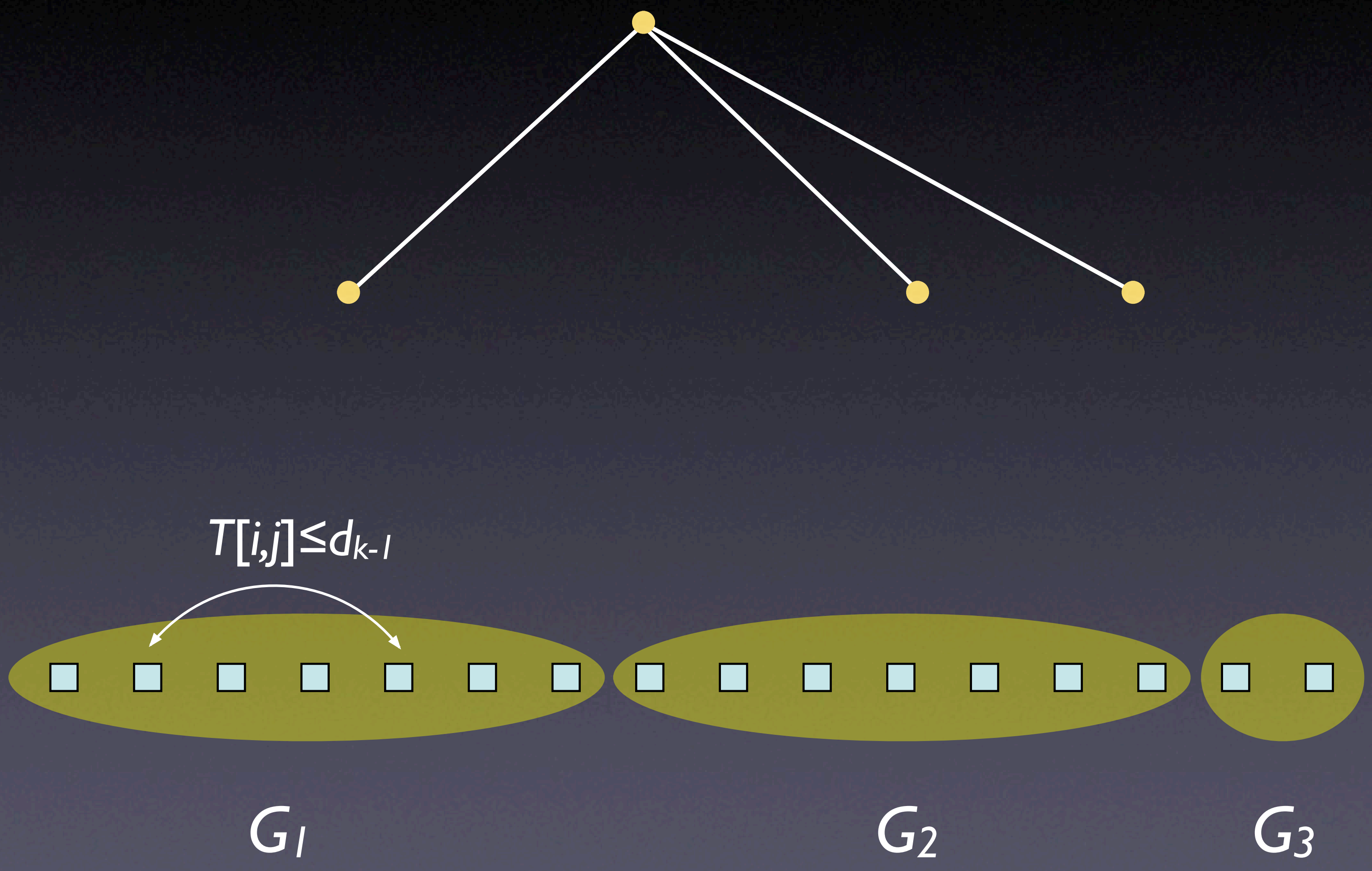




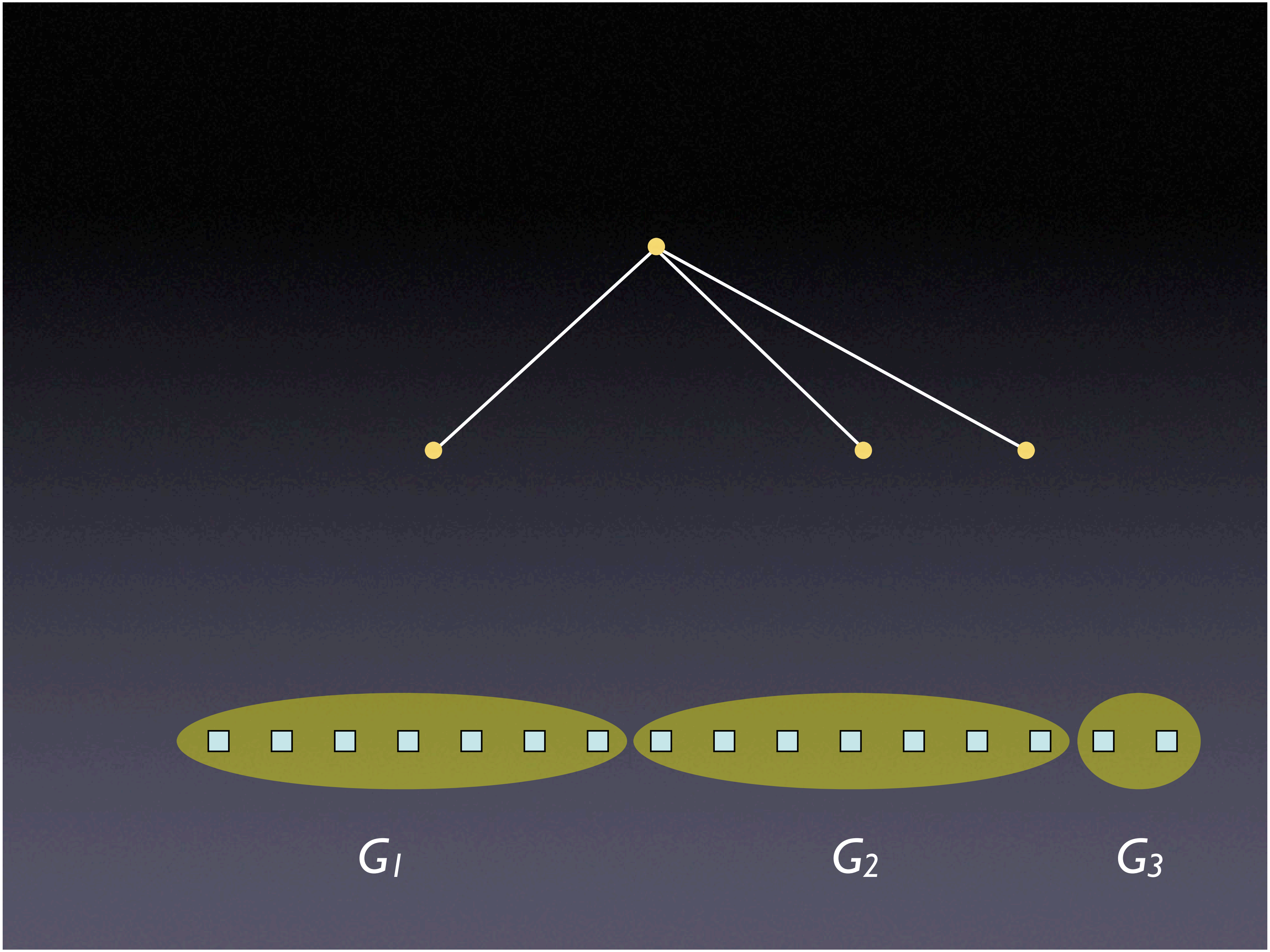
G_1
G_2
G_3



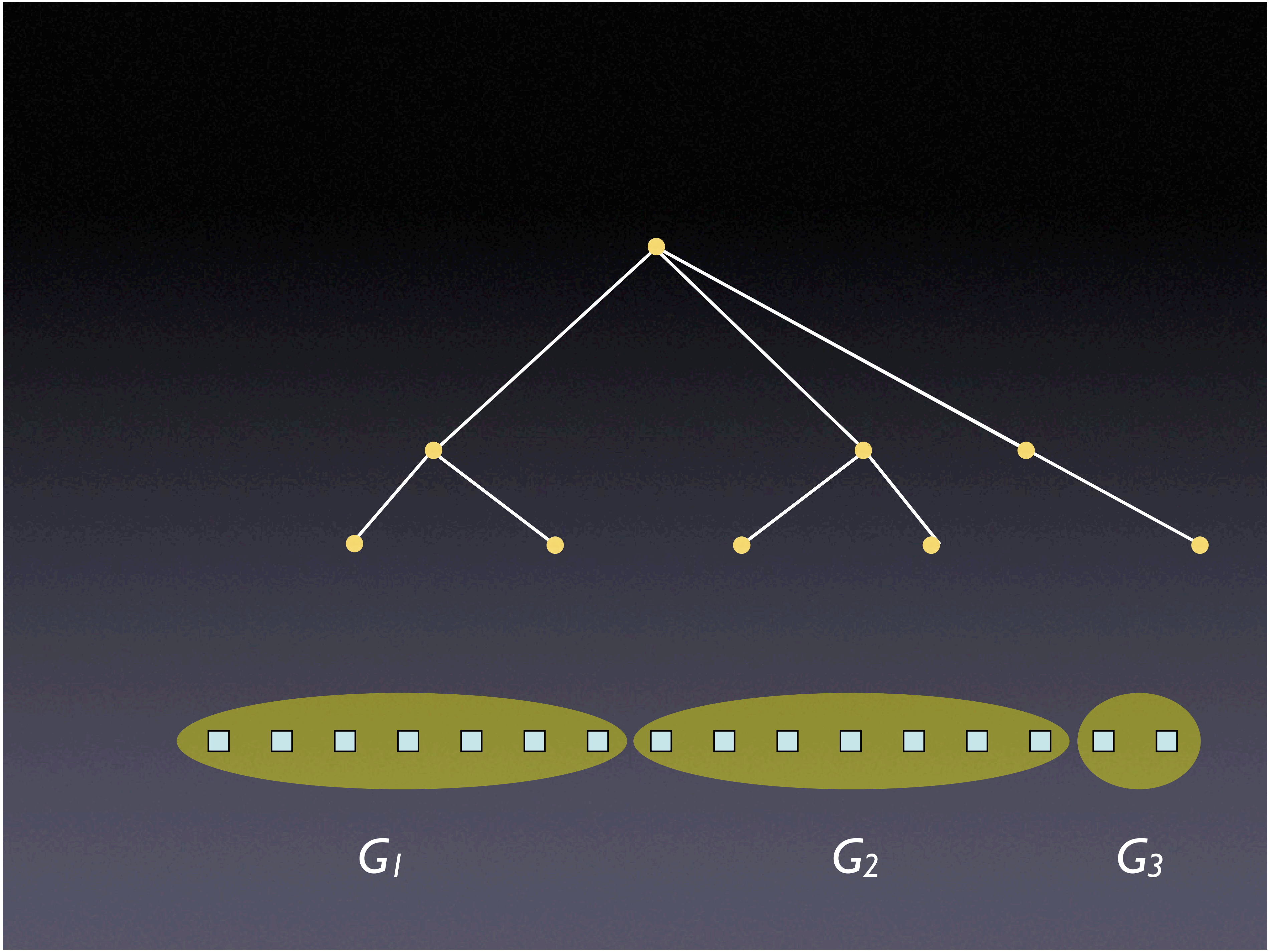
T[i,j]=d_k
G_1
G_2
G_3



$T[i,j] \leq d_{k-1}$
G_1
G_2
G_3



G_1
G_2
G_3


G_1
G_2
G_3

Correlation Clustering

- Input: Weighted (positive and negative) graph
- Output: A partitioning of nodes
- Goal: Minimize,

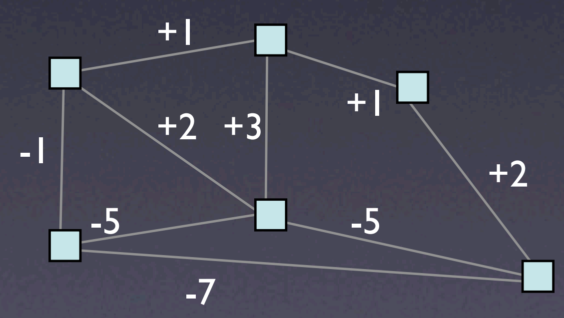
$$\sum_{e:w_e>0} (|w_e| \text{ if } e \text{ is split}) + \sum_{e:w_e<0} (|w_e| \text{ if } e \text{ is not split})$$

- O(log *n*) approximation [Charikar, Guruswami and Wirth '03]

Correlation Clustering

- Input: Weighted (positive and negative) graph
- Output: A partitioning of nodes
- Goal: Minimize,

$$\sum_{e:w_e>0} (|w_e| \text{ if } e \text{ is split}) + \sum_{e:w_e<0} (|w_e| \text{ if } e \text{ is not split})$$

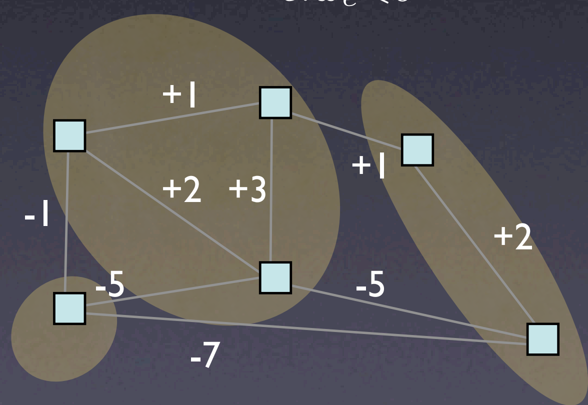


- O(log *n*) approximation [Charikar, Guruswami and Wirth '03]

Correlation Clustering

- Input: Weighted (positive and negative) graph
- Output: A partitioning of nodes
- Goal: Minimize,

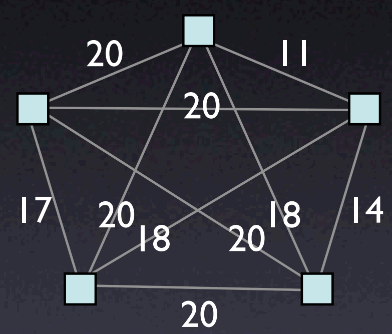
$$\sum_{e:w_e>0} (|w_e| \text{ if } e \text{ is split}) + \sum_{e:w_e<0} (|w_e| \text{ if } e \text{ is not split})$$



- O(log *n*) approximation [Charikar, Guruswami and Wirth '03]

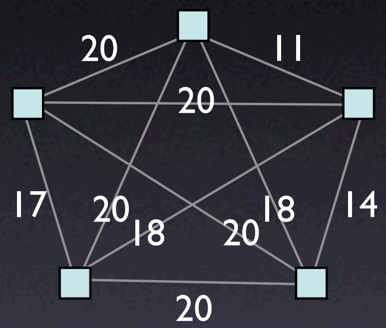
Using Correlation Clustering

Best-Fit Ultrametric Instance:



Using Correlation Clustering

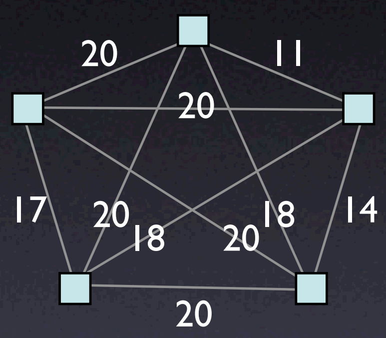
Best-Fit Ultrametric Instance:



Possible Splitting Distances: 20, 18, 17, 14, 11

Using Correlation Clustering

Best-Fit Ultrametric Instance:

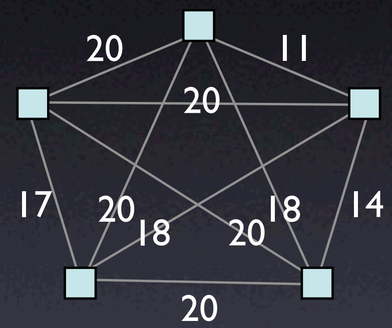


Possible Splitting Distances: 20, 18, 17, 14, 11

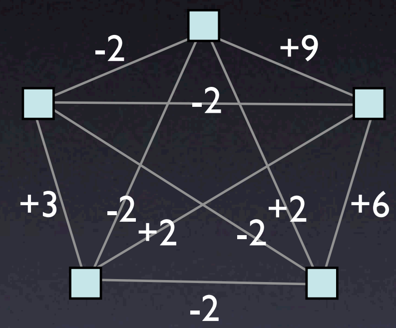
Top level clustering:
Increase some lengths to 20 and decrease some length 20 edges to 18

Using Correlation Clustering

Best-Fit Ultrametric Instance:

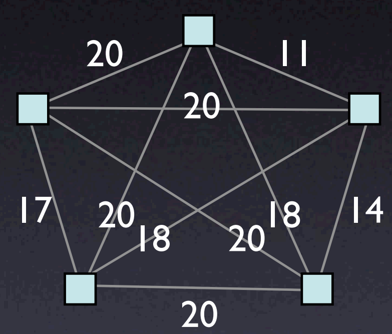


Correlation Clustering Instance:

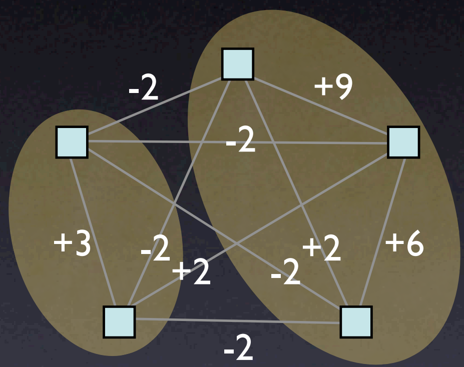


Using Correlation Clustering

Best-Fit Ultrametric Instance:

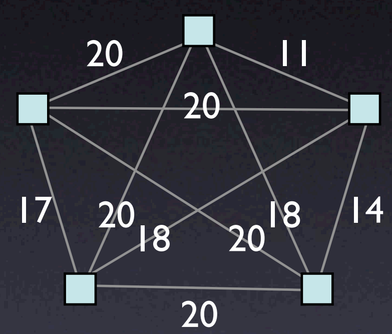


Correlation Clustering Instance:

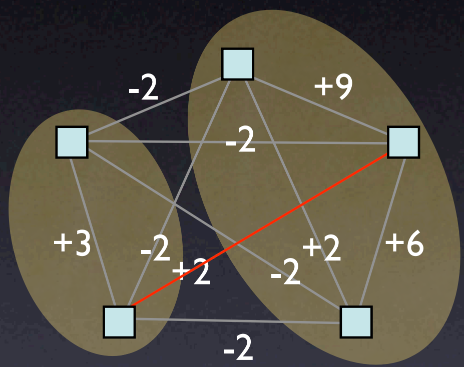


Using Correlation Clustering

Best-Fit Ultrametric Instance:

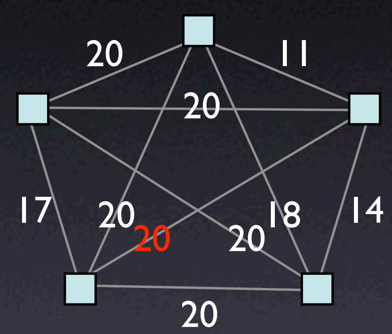


Correlation Clustering Instance:

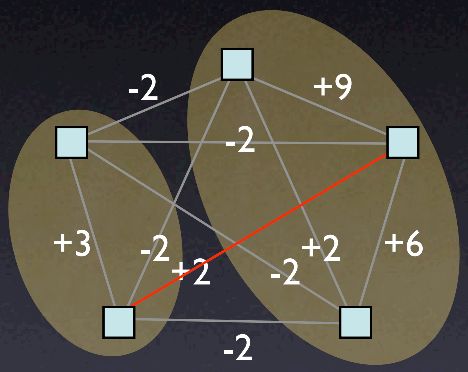


Using Correlation Clustering

Best-Fit Ultrametric Instance:

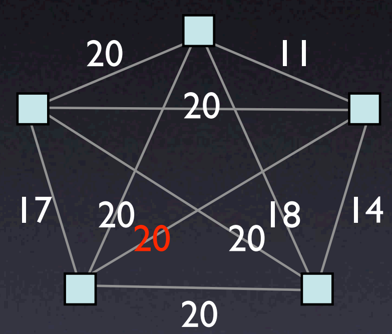


Correlation Clustering Instance:

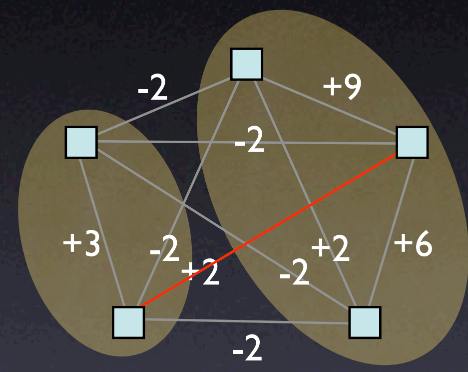


Using Correlation Clustering

Best-Fit Ultrametric Instance:



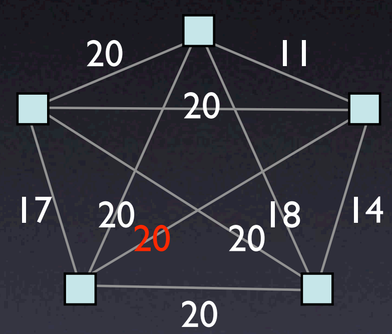
Correlation Clustering Instance:



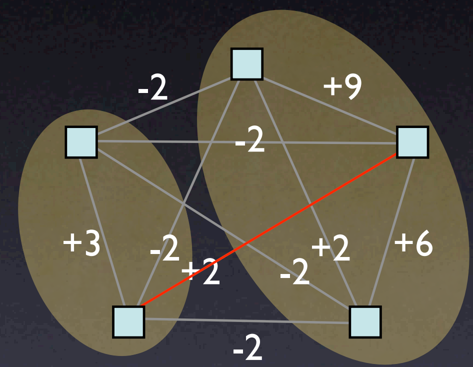
Cost of length changes = Cost of disagreements during clustering

Using Correlation Clustering

Best-Fit Ultrametric Instance:

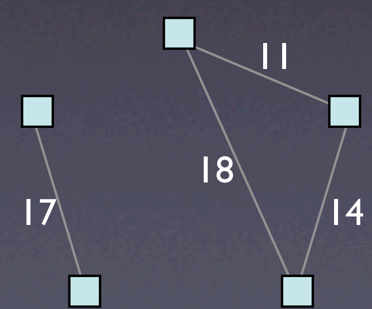


Correlation Clustering Instance:



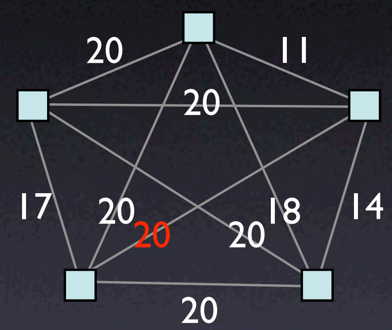
Cost of length changes = Cost of disagreements during clustering

Recurse:

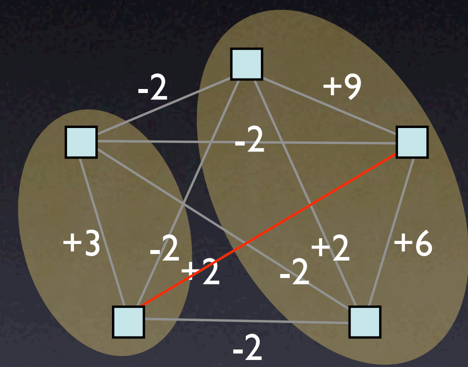


Using Correlation Clustering

Best-Fit Ultrametric Instance:

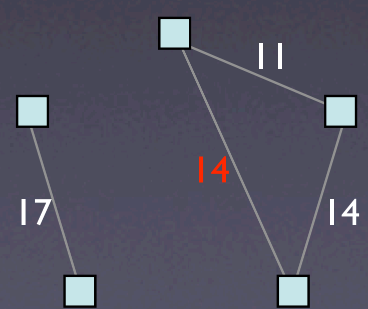


Correlation Clustering Instance:



Cost of length changes = Cost of disagreements during clustering

Recurse:



Analysis (Outline)

Analysis (Outline)

- Let OPT be cost of fit of best-fit tree (under L_1)

Analysis (Outline)

- Let OPT be cost of fit of best-fit tree (under L_1)
- In each of k steps:

  Cost of optimal clustering is ≤ OPT

  Cost of our clustering is ≤ O(log n) OPT

Analysis (Outline)

- Let OPT be cost of fit of best-fit tree (under L_l)
- In each of k steps:

  Cost of optimal clustering is ≤ OPT

  Cost of our clustering is ≤ O(log n) OPT
- Total cost of clusterings = $L_l(T, D) \le O(k \log n)$ OPT

Analysis (Outline)

- Let OPT be cost of fit of best-fit tree (under L_I)
- In each of k steps:
  Cost of optimal clustering is ≤ OPT
  Cost of our clustering is ≤ O(log n) OPT
- Total cost of clusterings = $L_I(T, D) \leq O(k \log n)$ OPT

Analysis (Outline)

- Let OPT be cost of fit of best-fit tree (under L_1)
- In each of k steps:

  Cost of optimal clustering is ≤ OPT

  Cost of our clustering is ≤ O(log n) OPT
- Total cost of clusterings = $L_1(T, D) \leq O(k \log n)$ OPT

- For L_p : seek to minimize $L_p^p\,(T, D)$

Analysis (Outline)

- Let OPT be cost of fit of best-fit tree (under L_1)
- In each of k steps:

  Cost of optimal clustering is ≤ OPT

  Cost of our clustering is ≤ O(log n) OPT
- Total cost of clusterings = $L_1(T, D) \leq O(k \log n)$ OPT

- For L_p : seek to minimize $L_p{}^p\,(T, D)$
- Similar analysis yields an $O(\log^2 n)$ approx under L_{rel}

Algorithm #2

Algorithm

- For $d = d_k$ to d_l:

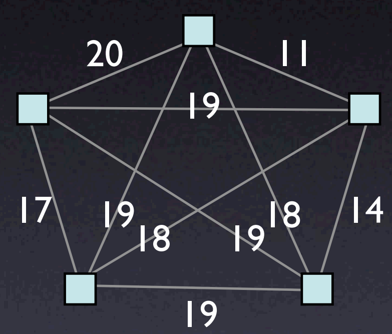
  Consider reducing maximum length to d and forcing a partition

  "Push-down-cost(d)" - cost of reducing each length ≥ d to d

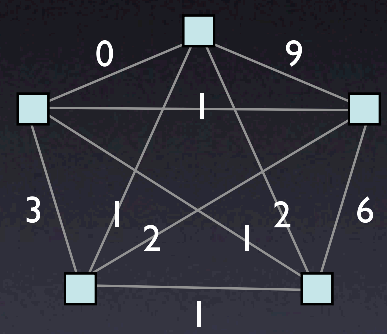
  "Cutting-cost(d)" - cost of increasing cut edge's length to d
- Split at d such that Push-down-cost(d)+ Cutting-cost(d)
- Recurse on each side of the cut

Using Minimum Cuts

Best-Fit Ultrametric Instance:



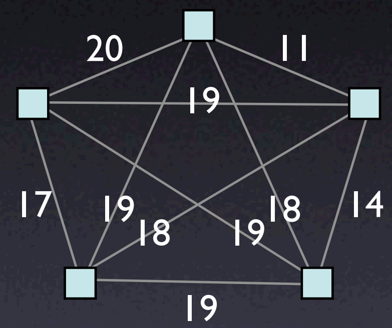
Minimum Cut Instance:



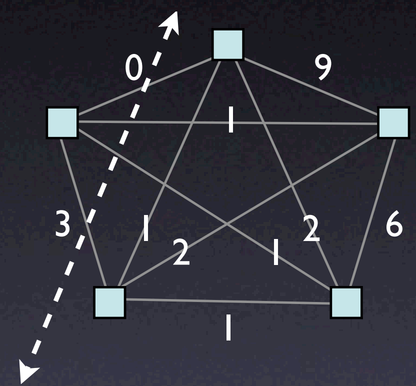
Split at 20:
Push-down cost = 0, Cut-cost = 3

Using Minimum Cuts

Best-Fit Ultrametric Instance:



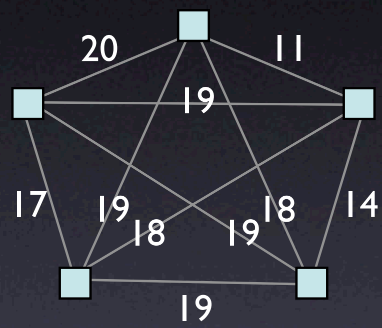
Minimum Cut Instance:



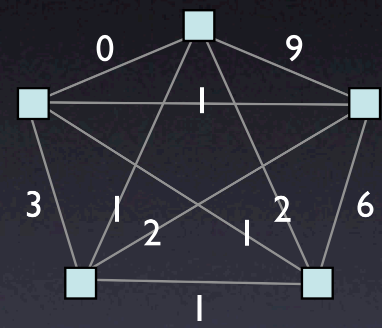
Split at 20:
Push-down cost = 0, Cut-cost = 3

Using Minimum Cuts

Best-Fit Ultrametric Instance:



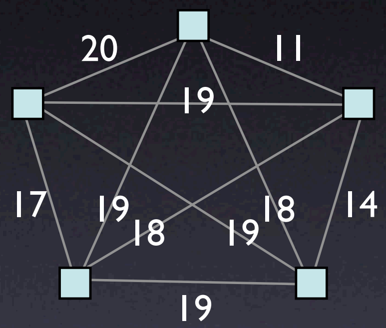
Minimum Cut Instance:



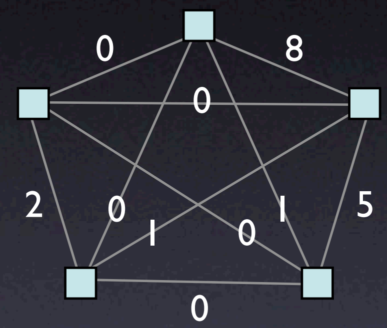
Split at 20:
Push-down cost = 0, Cut-cost = 3

Using Minimum Cuts

Best-Fit Ultrametric Instance:



Minimum Cut Instance:

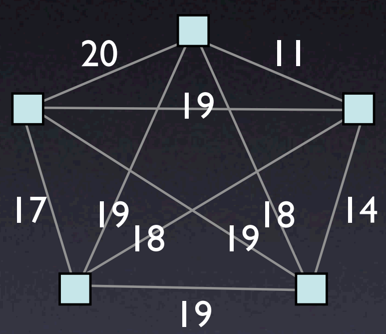


Split at 20:
Push-down cost = 0, Cut-cost = 3

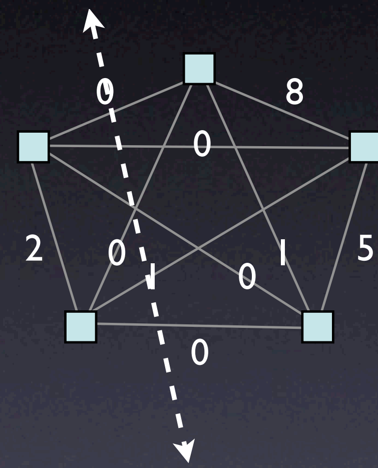
Split at 19:
Push-down cost = 1, Cut-cost =1

Using Minimum Cuts

Best-Fit Ultrametric Instance:



Minimum Cut Instance:

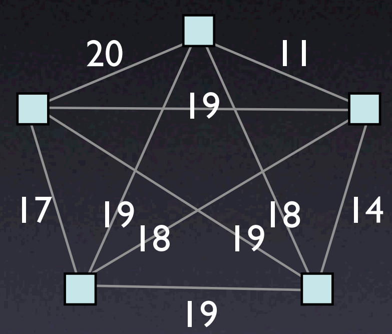


Split at 20:
Push-down cost = 0, Cut-cost = 3

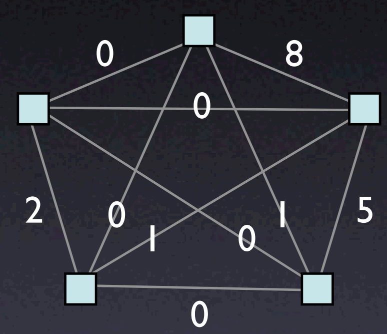
Split at 19:
Push-down cost = 1, Cut-cost =1

Using Minimum Cuts

Best-Fit Ultrametric Instance:



Minimum Cut Instance:

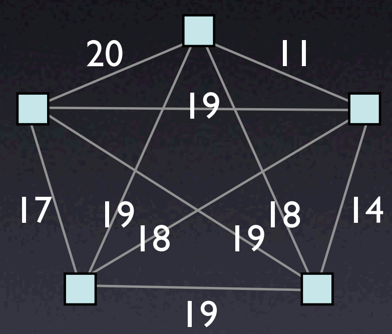


Split at 20:
Push-down cost = 0, Cut-cost = 3

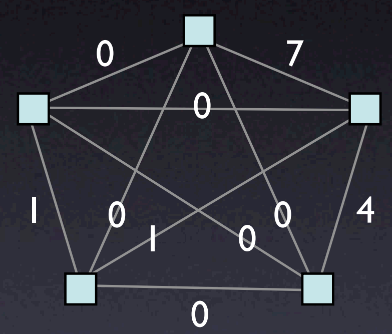
Split at 19:
Push-down cost = 1, Cut-cost =1

Using Minimum Cuts

Best-Fit Ultrametric Instance:



Minimum Cut Instance:



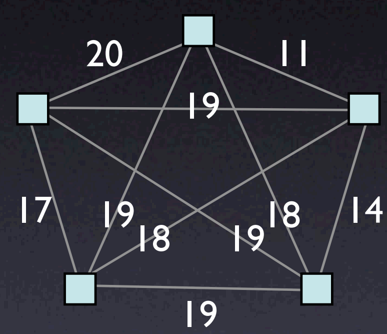
Split at 20:
Push-down cost = 0, Cut-cost = 3

Split at 19:
Push-down cost = 1, Cut-cost =1

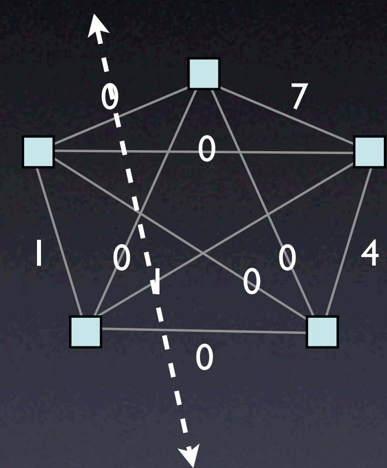
Split at 18:
Push-down cost = 5, Cut-cost = 0

Using Minimum Cuts

Best-Fit Ultrametric Instance:



Minimum Cut Instance:



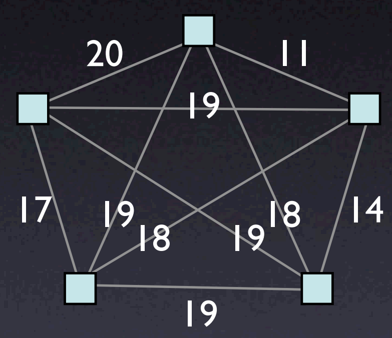
Split at 20:
Push-down cost = 0, Cut-cost = 3

Split at 19:
Push-down cost = 1, Cut-cost =1

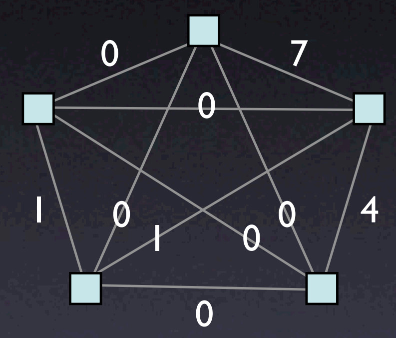
Split at 18:
Push-down cost = 5, Cut-cost = 0

Using Minimum Cuts

Best-Fit Ultrametric Instance:



Minimum Cut Instance:



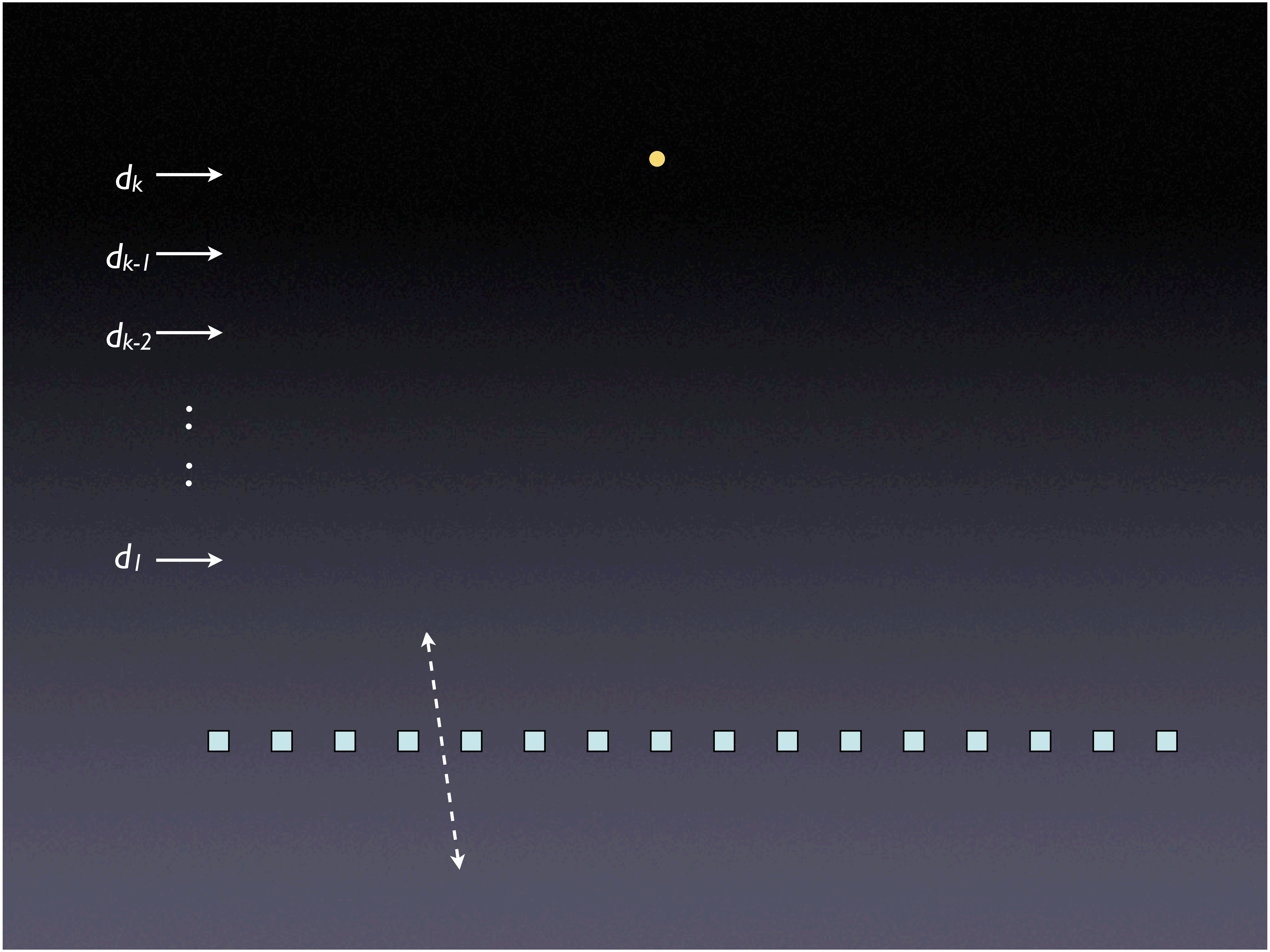
Split at 20:
Push-down cost = 0, Cut-cost = 3

Split at 19:
Push-down cost = 1, Cut-cost =1

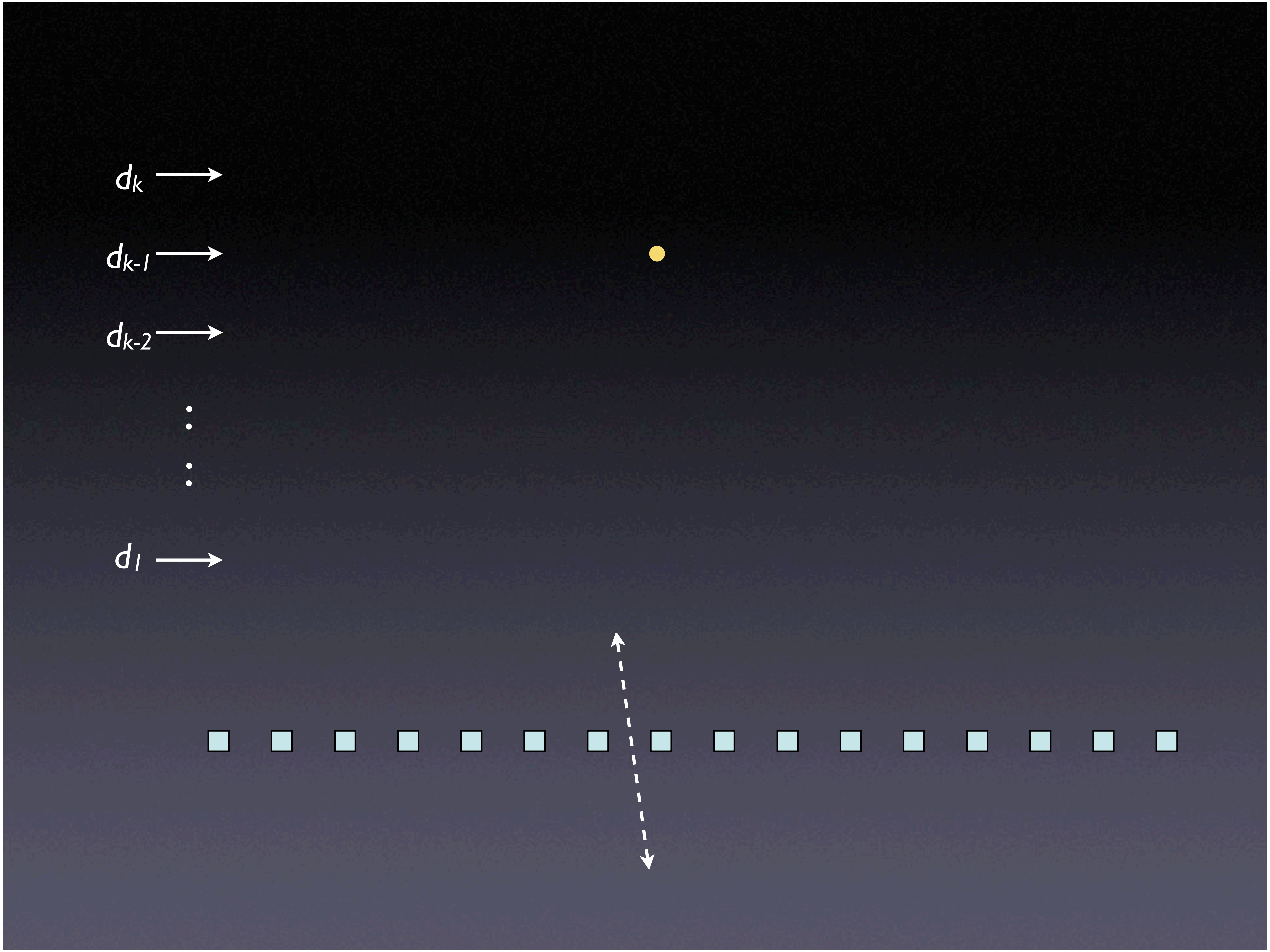
Split at 18:
Push-down cost = 5, Cut-cost = 0


d_k
d_{k-1}
d_{k-2}
d_1

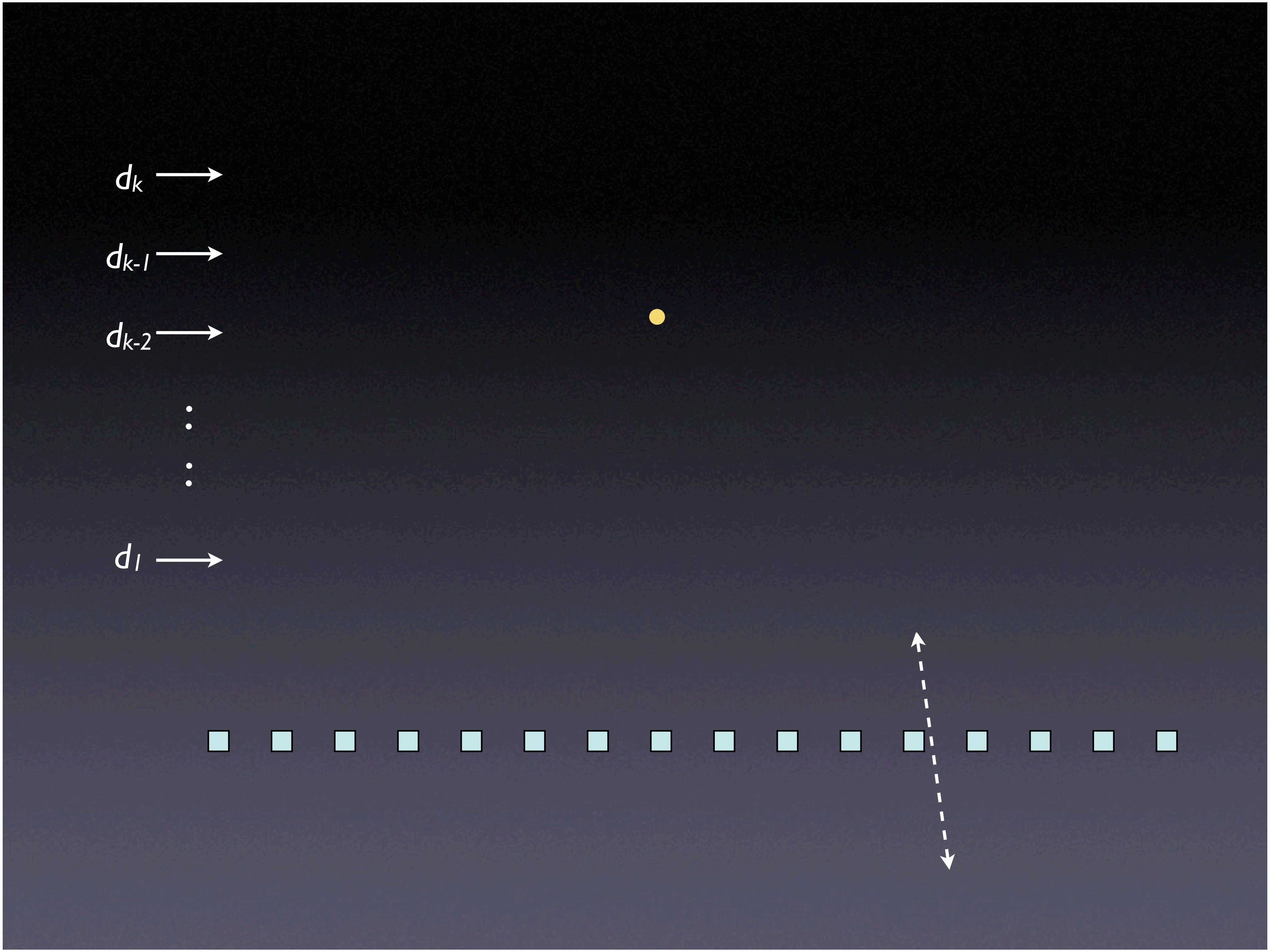
d_k
d_{k-1}
d_{k-2}
d_1


d_k
d_{k-1}
d_{k-2}
d_1

d_k
d_{k-1}
d_{k-2}
d_1

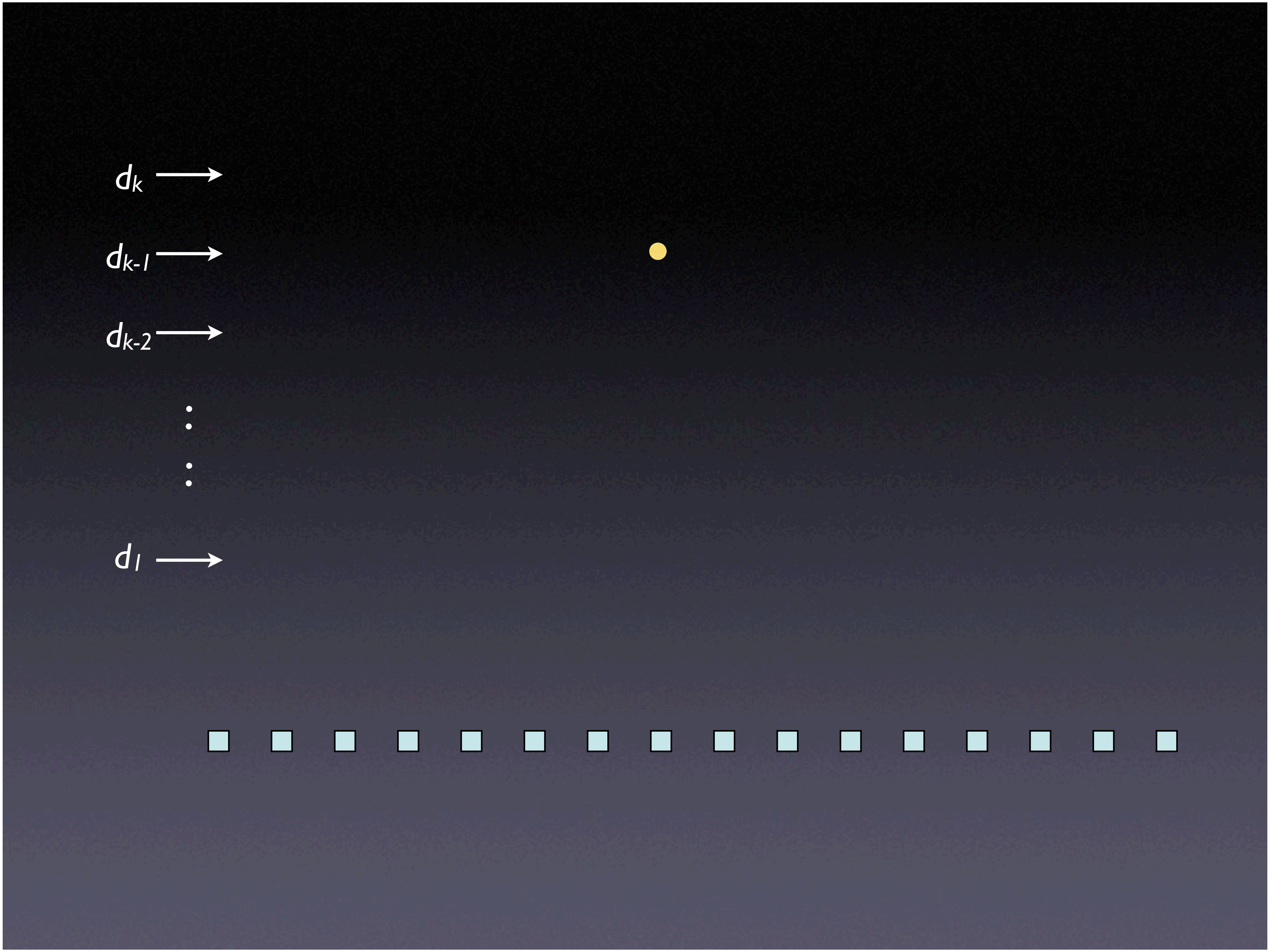

d_k
d_{k-1}
d_{k-2}
d_1


d_k
d_{k-1}
d_{k-2}
d_1

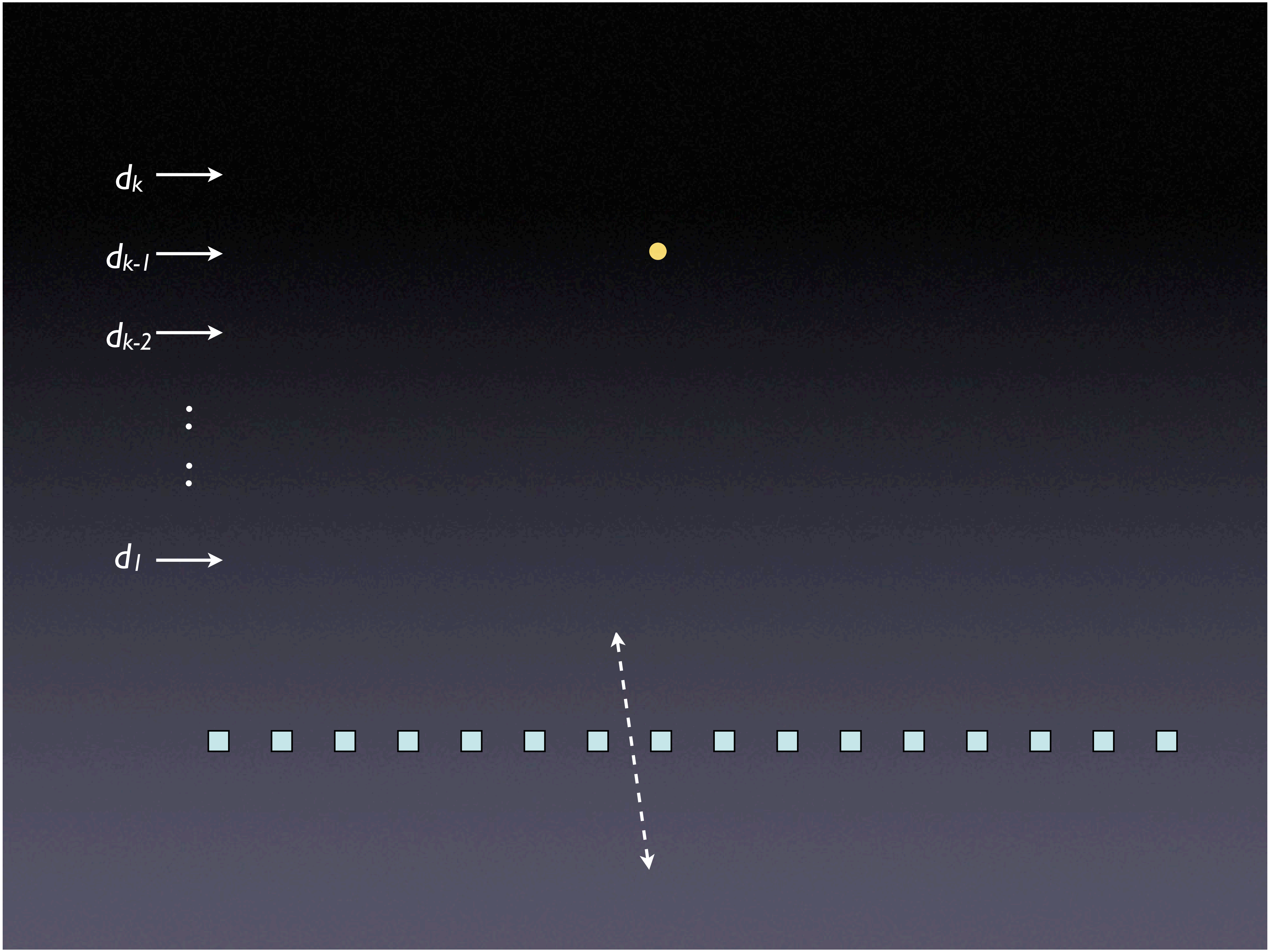


d_k
d_{k-1}
d_{k-2}
d_1

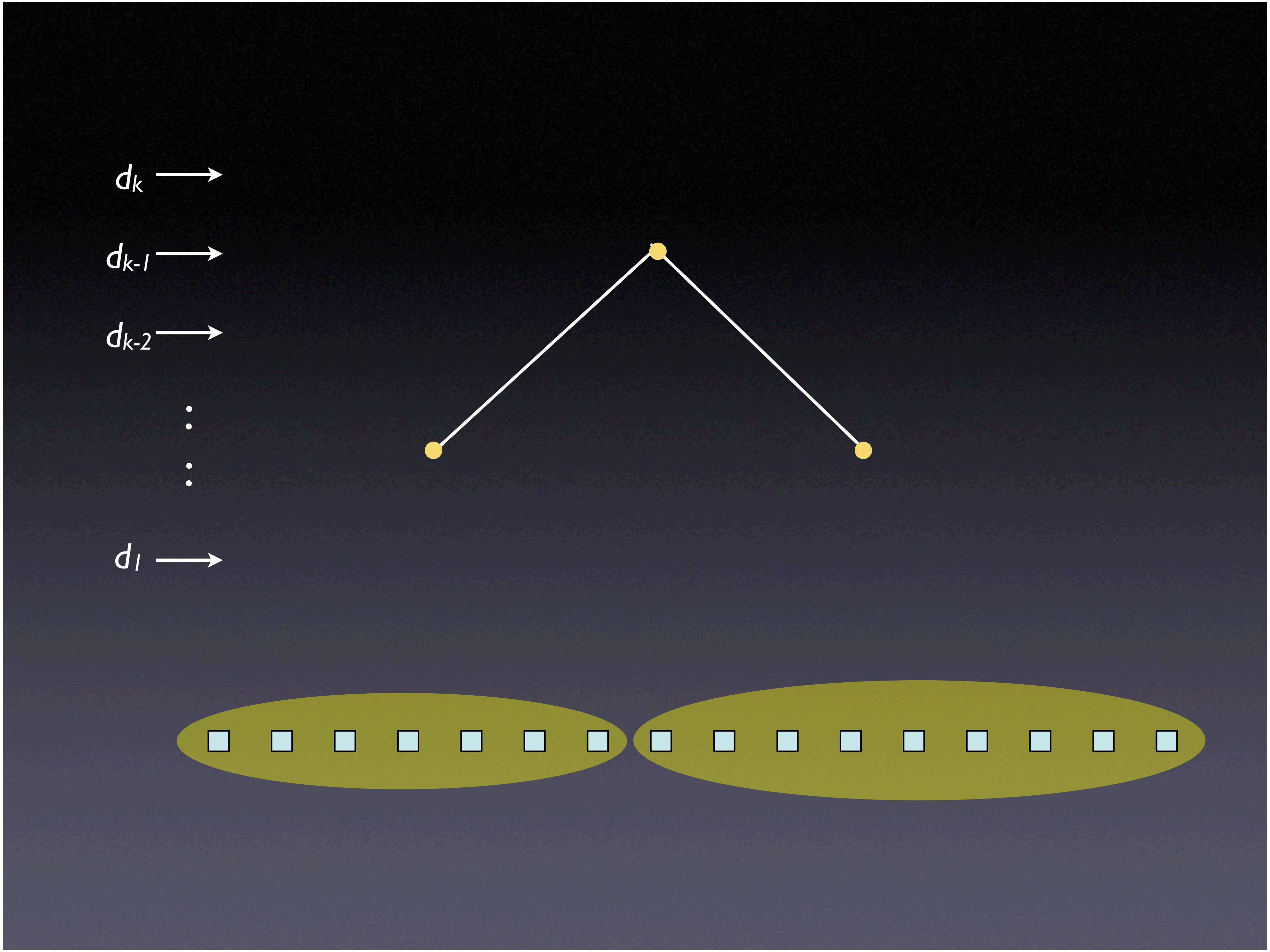

d_k
d_{k-1}
d_{k-2}
d_1



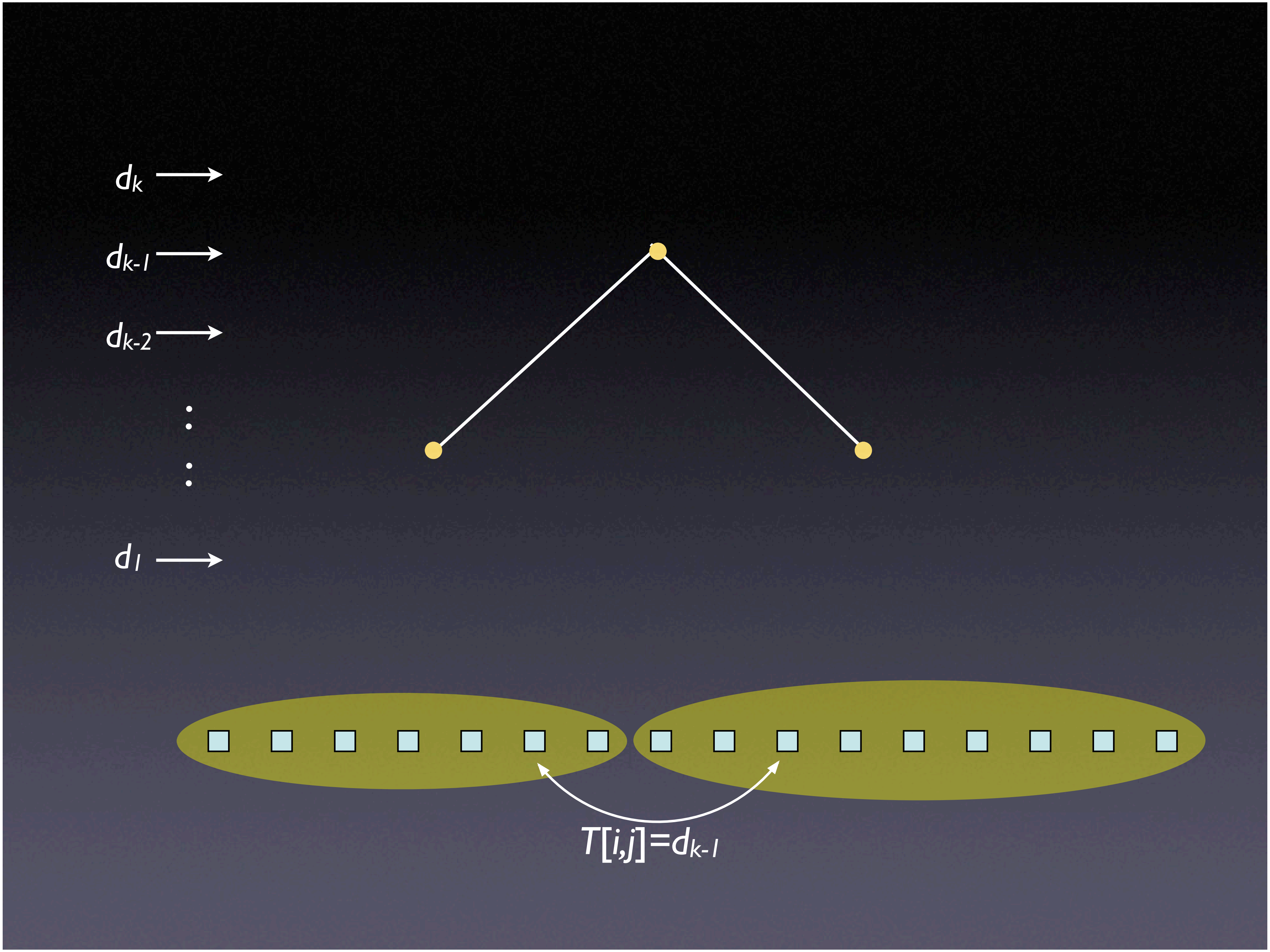
d_k
d_{k-1}
d_{k-2}
d_1



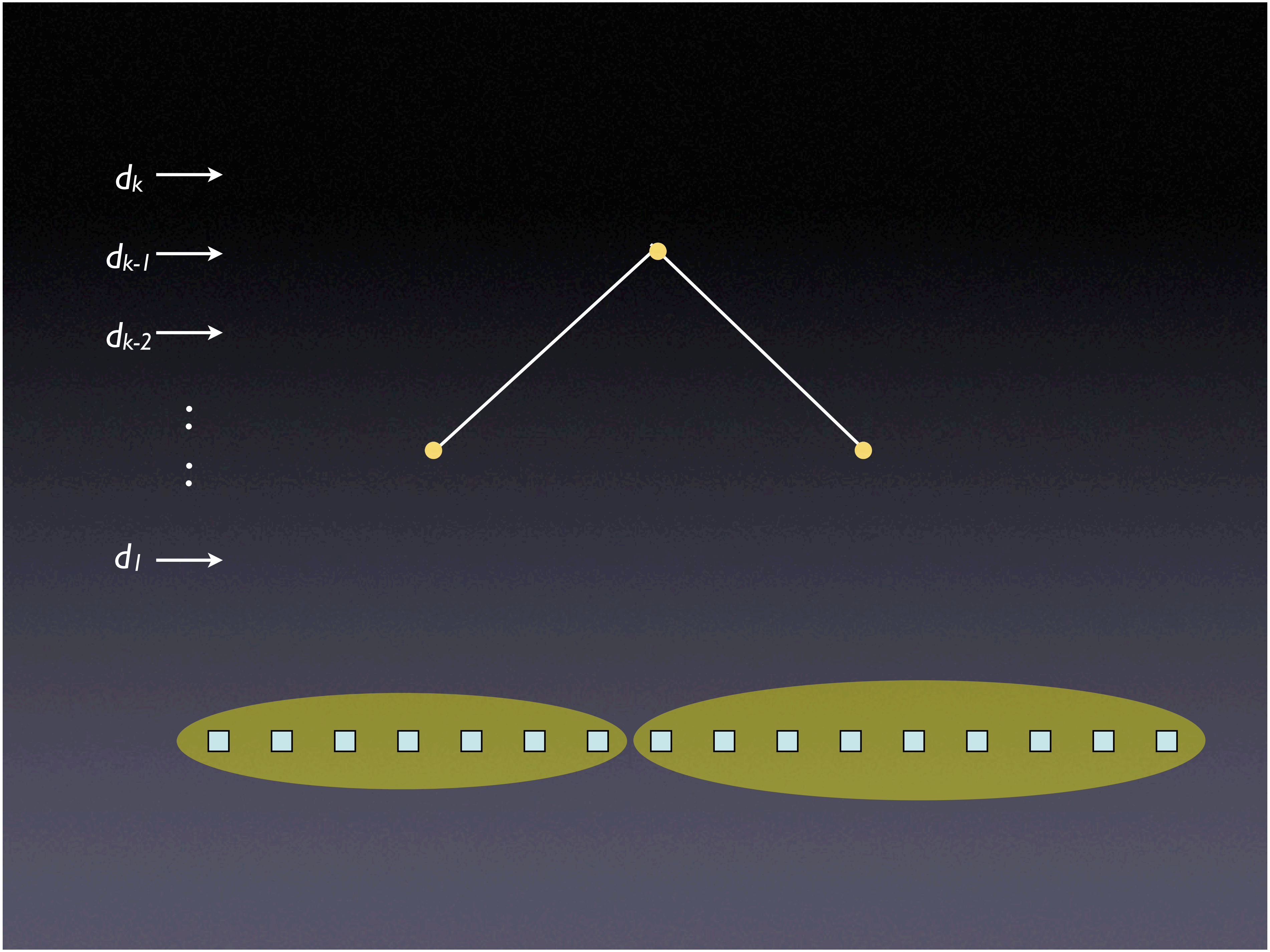
d_k
d_{k-1}
d_{k-2}
d_1



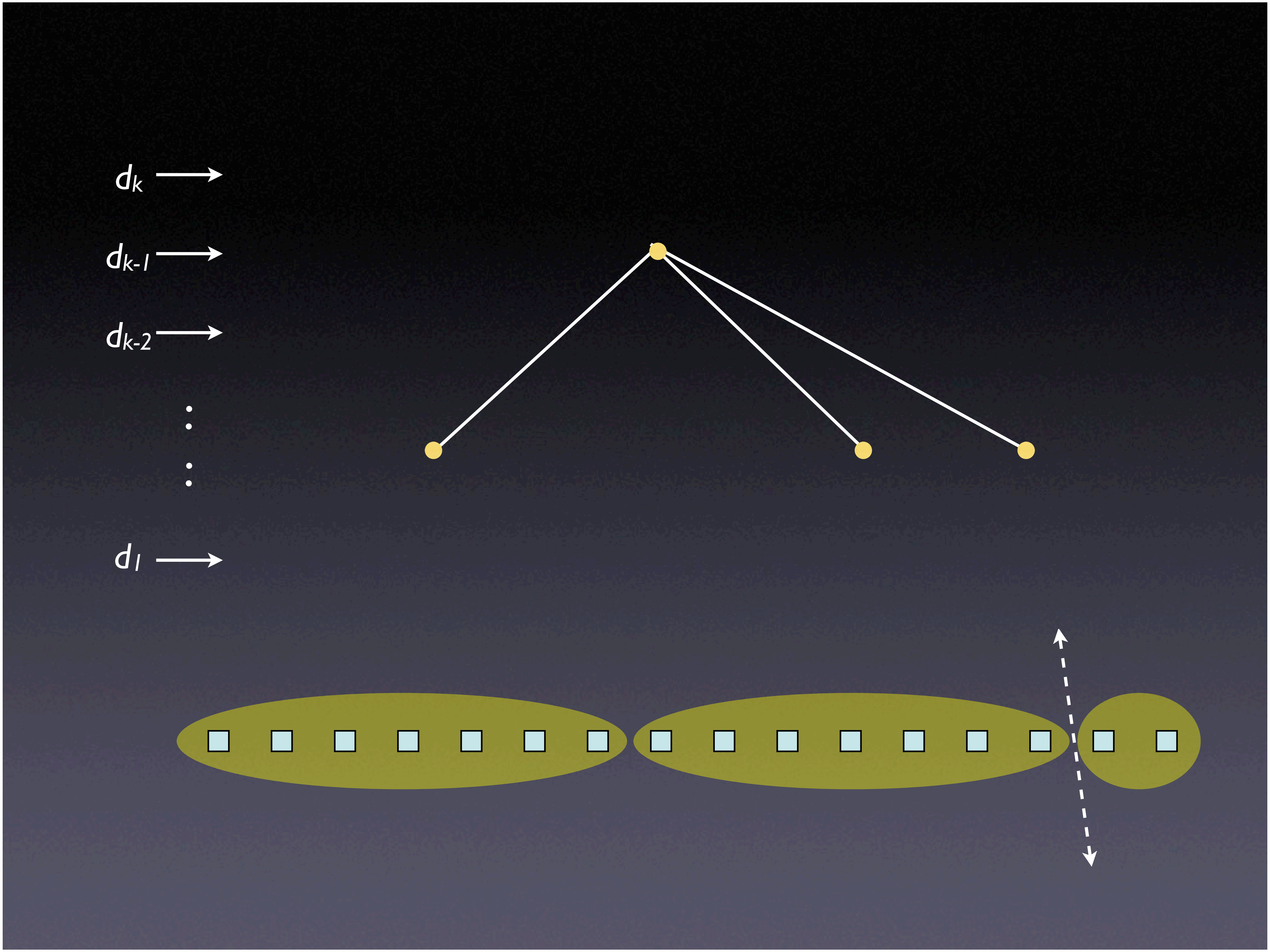
d_k
d_{k-1}
d_{k-2}
d_1



d_k
d_{k-1}
d_{k-2}
d_1
$T[i,j]=d_{k-1}$



d_k
d_{k-1}
d_{k-2}
d_1



d_k
d_{k-1}
d_{k-2}
d_1

Analysis (Outline)

- There are at most n cuts to be found
- For each cut:

  $\min_d$ Push-down-cost(d) + Cutting-cost(d) $\leq$ OPT

- Total Cost = $L_p(T,D)$

Extending to Trees

Extending to Trees

- Theorem [Agarwala, Bafna, Farach, Paterson, Thorup '99]:
  An α-approx to the optimal "a-restricted ultrametric" (under L_p) can be used to construct an 3α-approx to the optimal tree metric under (under L_p).

Extending to Trees

- Theorem [Agarwala, Bafna, Farach, Paterson, Thorup '99]:

  An α-approx to the optimal "*a*-restricted ultrametric" (under L_p) can be used to construct an 3α-approx to the optimal tree metric under (under L_p).

- Definition: An *a*-restricted ultrametric satisfies:

  For all i, $T[a,i] = 2\mu$

  For all i,j, $2\mu \geq T[i,j] \geq 2\,(\mu - \min\,(D[a,i], D[a,j]))$

  where $\mu = \max_i D[a,i\,]$

Conclusions

- Best-fit Tree or Ultrametric:

  L_p: $O(\min(n, k \log n))^{1/p}$ approximation where k is the number of distinct distances in D

  L_{rel}: $O(\log^2 n)$ approximation

- Best-fit Tree:

  L_p: $n^{1/p}$ approximation

Conclusions

- Best-fit Tree or Ultrametric:

  L_p: $O(\min(n, k \log n))^{1/p}$ approximation where k is the number of distinct distances in D

  L_{rel}: $O(\log^2 n)$ approximation

- Best-fit Tree:

  L_p: $n^{1/p}$ approximation

Late Breaking News: Upcoming FOCS paper by Ailon and Charikar has improved results!


Thanks.