Optimization over Tree Structured Objects

Presented by: Burcu Aydin

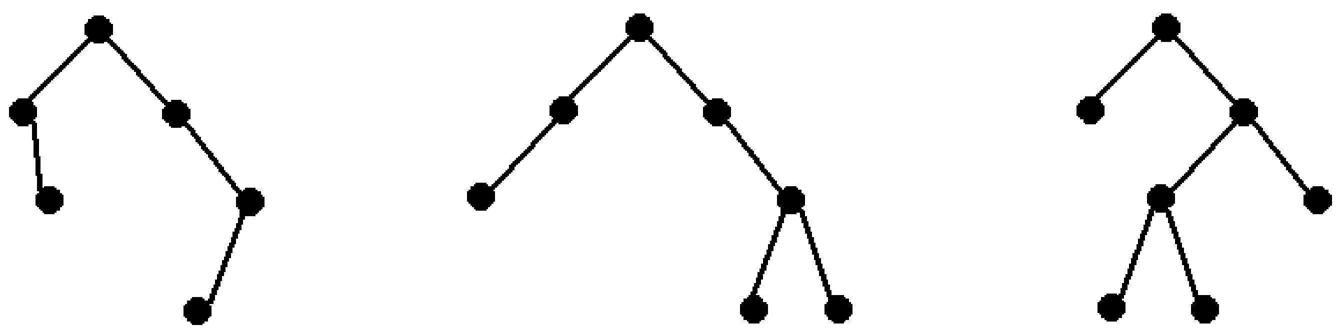
Advisors: Prof. Pataki, Prof. Marron

Treeline Problem

- Optimization point of view
- Start from first Principal Component

Data:

- A set of binary trees (Or, set of points in the binary tree space):
- $T = \{t_1, t_2, \ldots, t_n\}$
- An example data set:



Objective:

- Finding best approximating *treeline*
- Need to define these first!

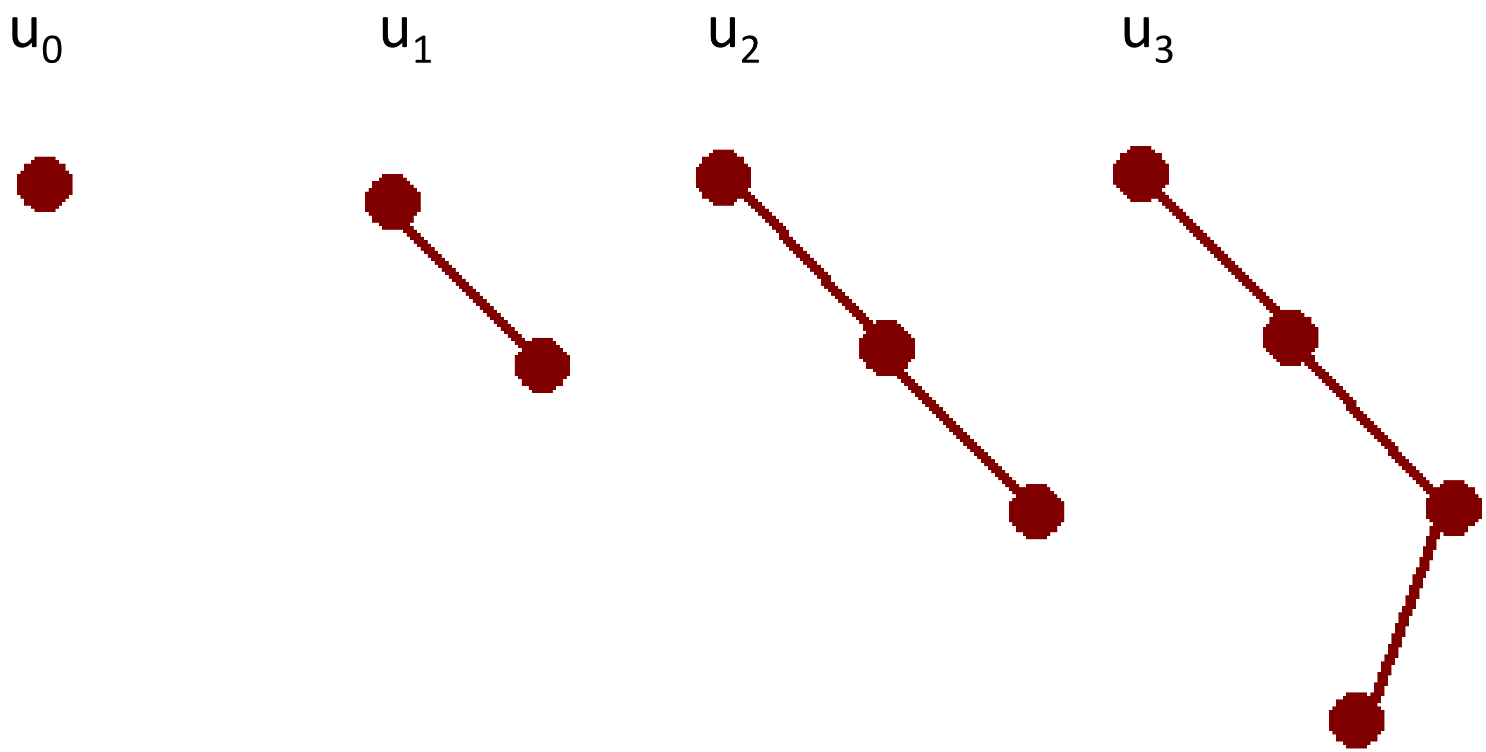
A treeline is…

- A set of trees, L, such that:
  - $L=\{u_0,u_1,u_2,\ldots,u_m\}$
  - u_i can be obtained by adding a single node (v_i) to u_{i-1}
  - v_{i+1} is a child of vi

- Also, we will call the set of nodes hanged to u_0 as Path(L), path of treeline L:

$$\mathrm{Path}(L) = u_m/u_0$$

An example treeline:



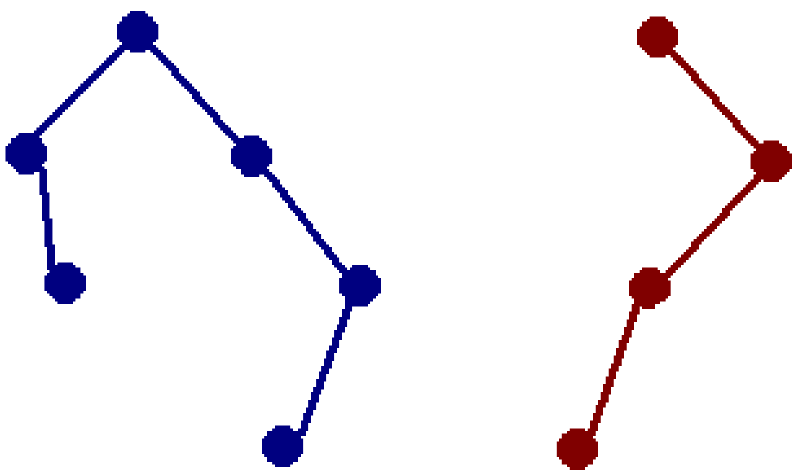
Projection..

- of a data tree t_i onto treeline L is the closest point on L to the data tree:

$$P_L(t_i) = \arg\min_{u_j \in L} d(t_i, u_j)$$

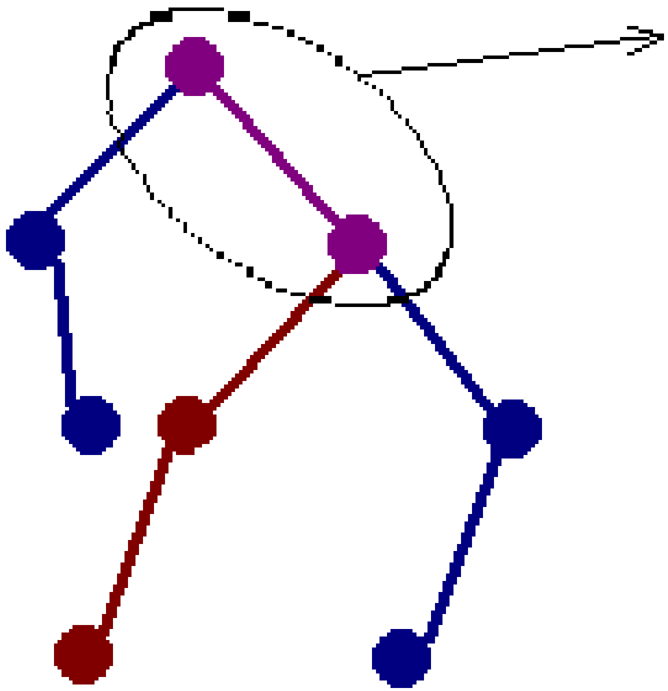
But what is Distance?

- The number of nodes in the symmetric difference of two trees.
- An example:



An example

- The trees drawn ‘on top of each other’:



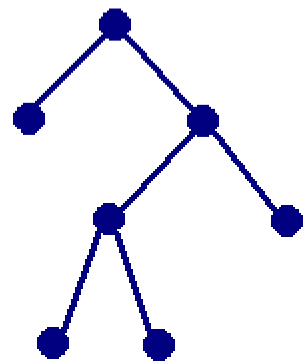
Common nodes: 2

Nodes only in blue tree: 4
Nodes only in red tree: 2

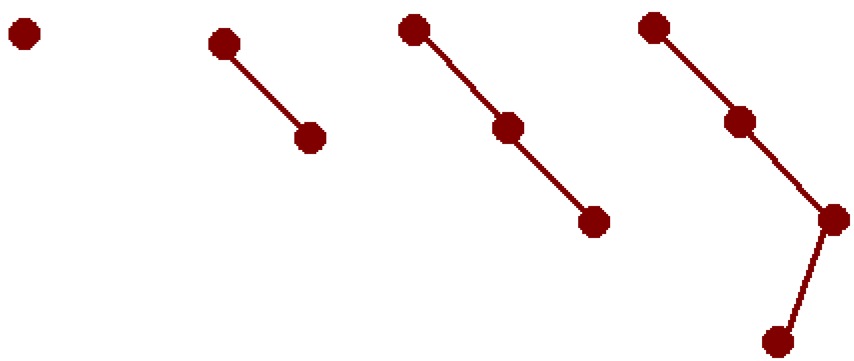
So, distance: 4+2=6

An example of projection

- Suppose we have the data tree:



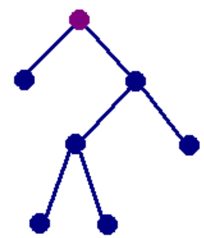
- And the treeline:



Which point on the treeline is the closest to the data tree?

An example of projection

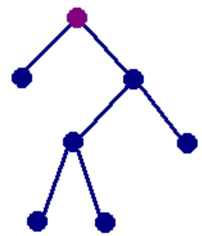
u_0:



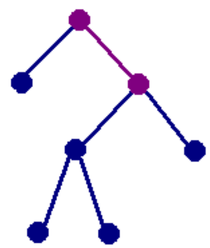
$d(u_0,t) = 6$

An example of projection

u_0:



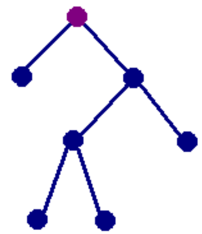
$d(u_0,t) = 6$

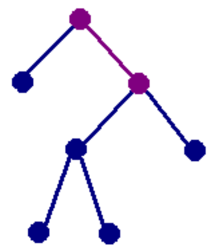


u_1:

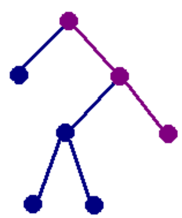
$d(u_1,t)=5$

An example of projection

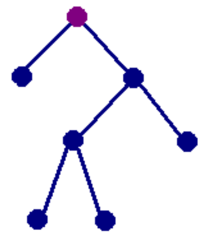
u_0:  $d(u_0,t) = 6$

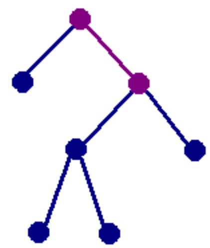


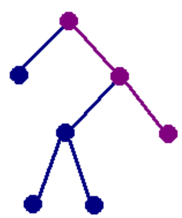
u_1: $d(u_1,t)=5$

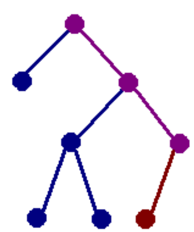
u_2:  $d(u_2,t)=4$

An example of projection

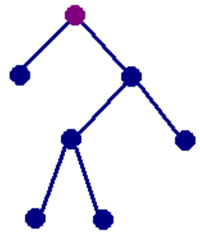
u_0:  $d(u_0,t) = 6$

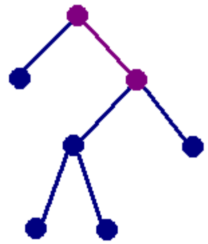
u_1:  $d(u_1,t)=5$

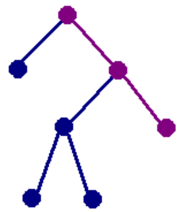
u_2:  $d(u_2,t)=4$

u_3:  $d(u_3,t)=5$

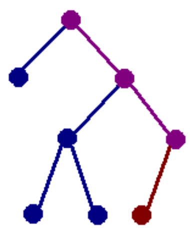
An example of projection

u_0:  $d(u_0,t) = 6$

u_1:  $d(u_1,t)=5$

u_2:  $d(u_2,t)=4$

Closest!

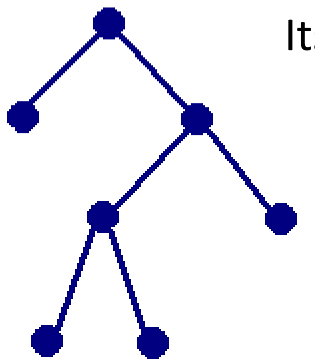
u_3:  $d(u_3,t)=5$

An example of projection

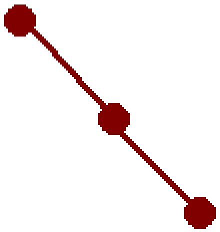
- So, $P_L(t)=u_2$ in this case

- Observation:
  - Projection onto the treeline is u_0 combined with the members of P(L) that are in the data tree:

The data tree:



Its projection:



Best Approximation

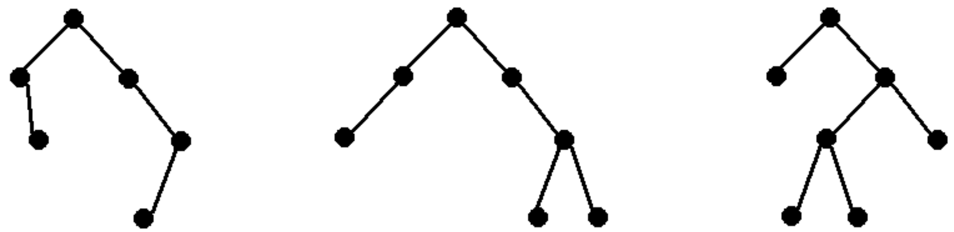
- The treeline that gives the smallest sum of distances:

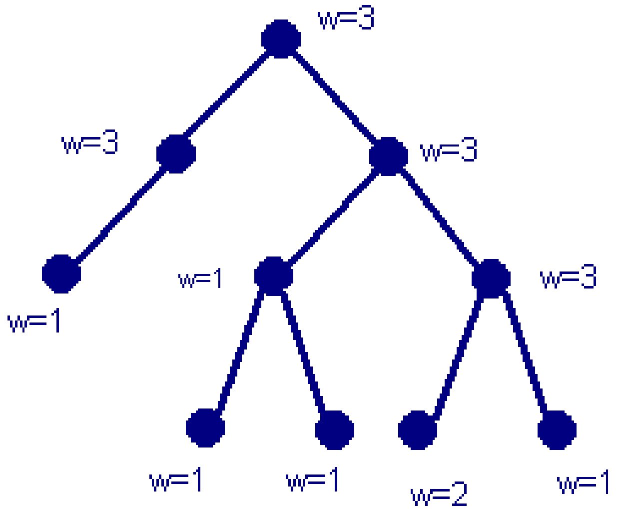
$$L^* = \arg\min_L \{ \sum_{t_i \in T} \min_{u_j \in L} d(t_i, u_j) \}$$

Support Tree

- The union of all trees in data set T.
- Consists of the nodes that exist in T and their “weights”.
- Weight of a node, w(v,T) is the number of trees it occurs in the data set.

On the example:

- Data trees: 

- Support tree: 

How the method works

- First, write out the objective:

$$D_{u_0} = \min_{L \in L_{u_0}} \{\sum_{t_i \in T} d(t_i, P_L(t_i))\}$$

How the method works

- In terms of the path:

$$\min_{L \in L_{u_0}} \sum_{t_i \in T} d(t_i, u_0 \cup P(L))$$

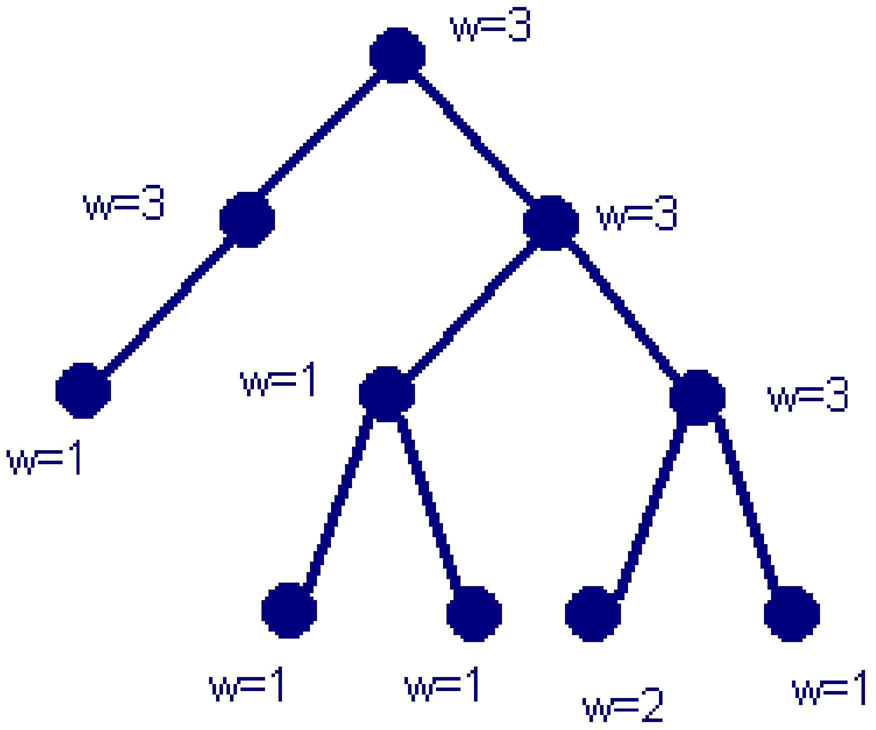
- After intermediate steps:

$$= \sum_{t_i \in T} d(t_i, u_0) - \max_{L \in L_{u_0}} \sum_{v \in P(L)} w(v, T)$$

So, we need:

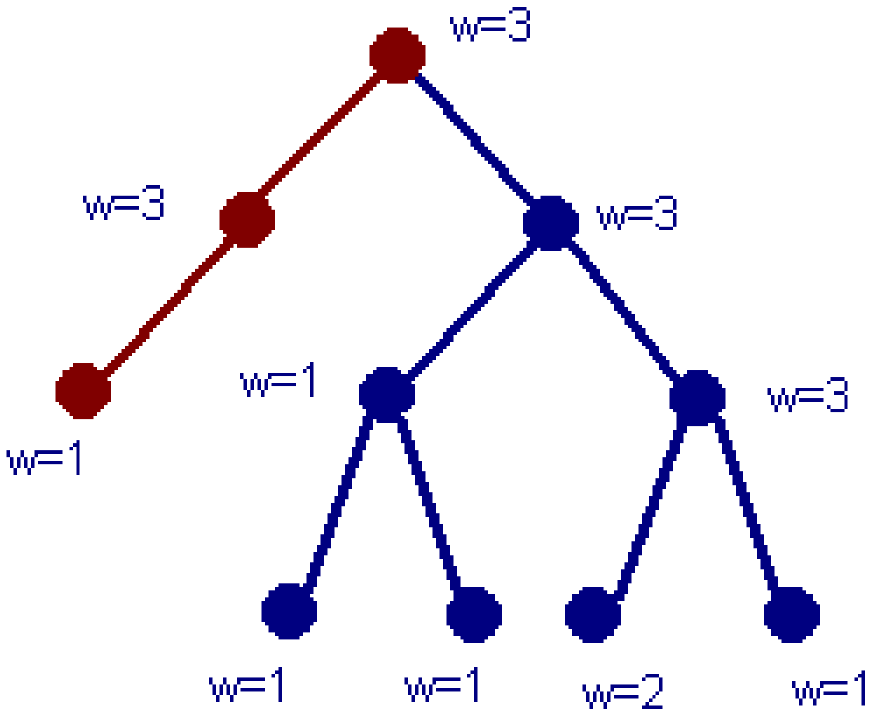
- The path with largest sum of weights..
- How many paths are there to check?
- Constructing the Support Tree takes O(n) time
- Finding the path with largest weight sum on it is another O(n)

On the example:

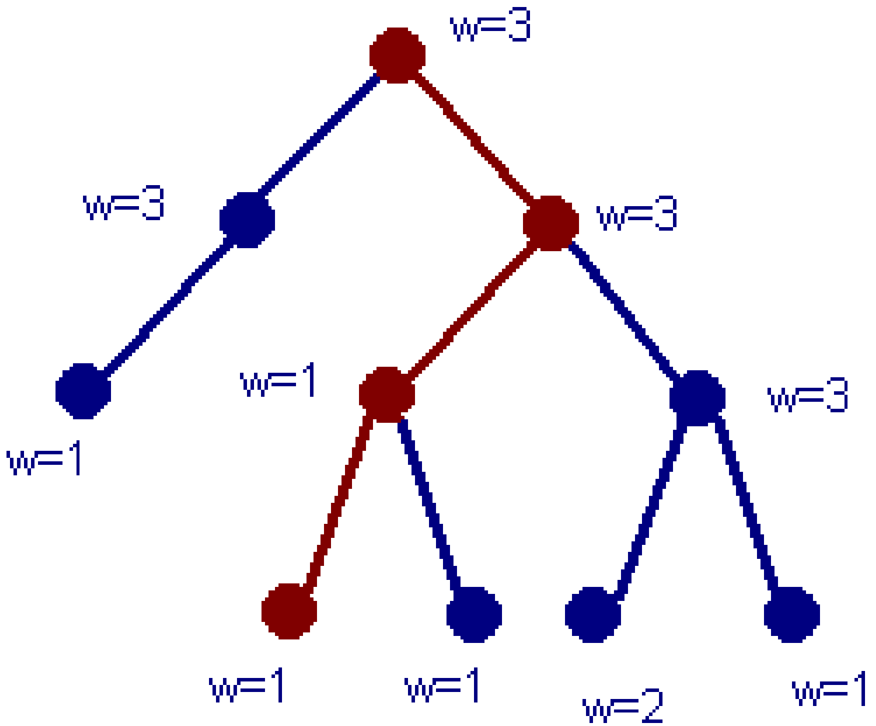


On the example:

First Path:
$\sum w = 7$

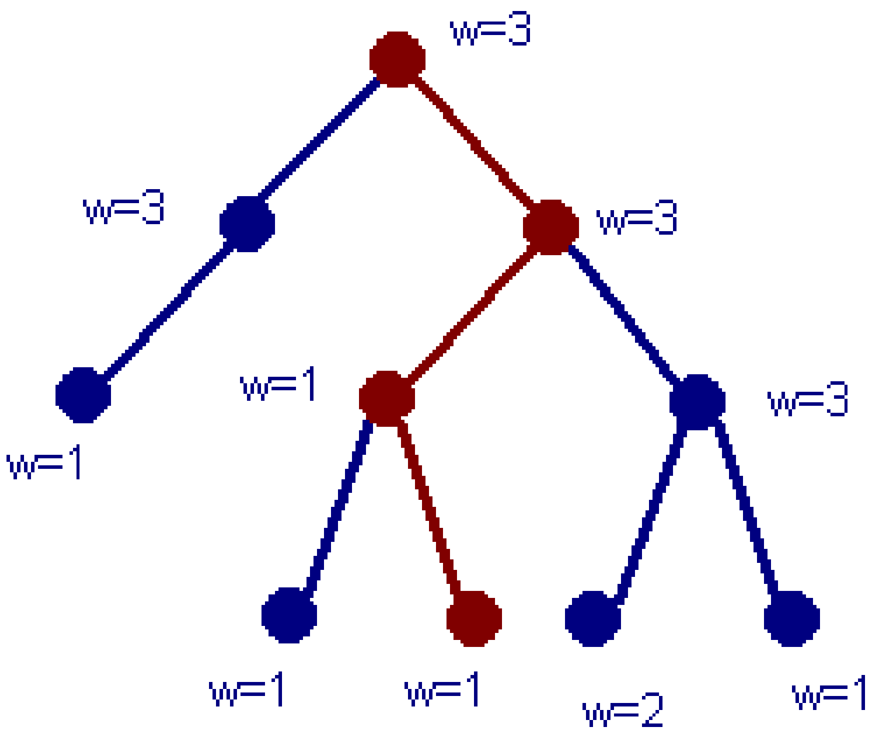


On the example:



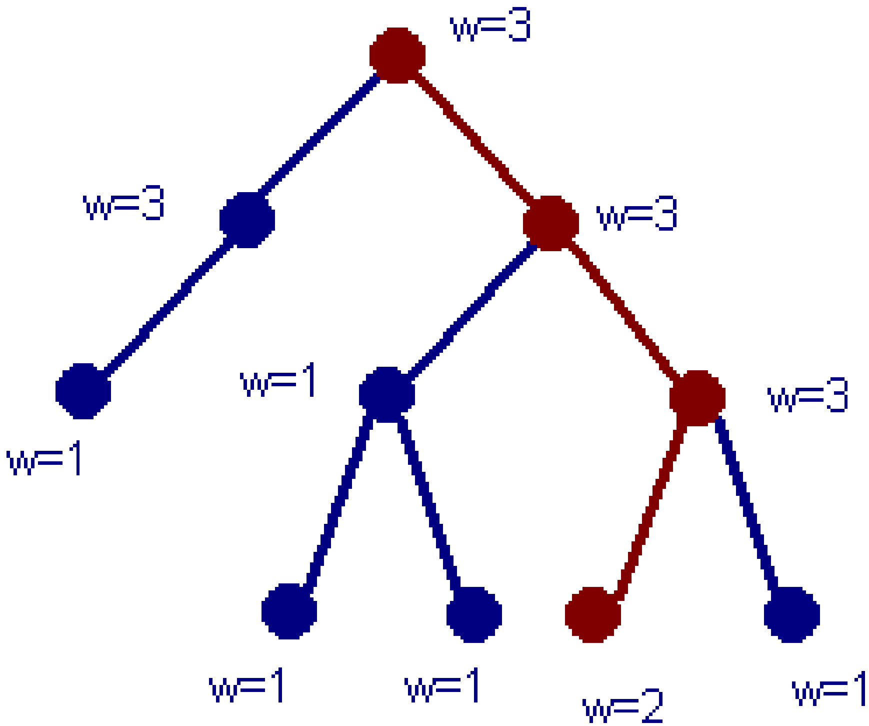
Second Path:
$\sum w = 8$

On the example:



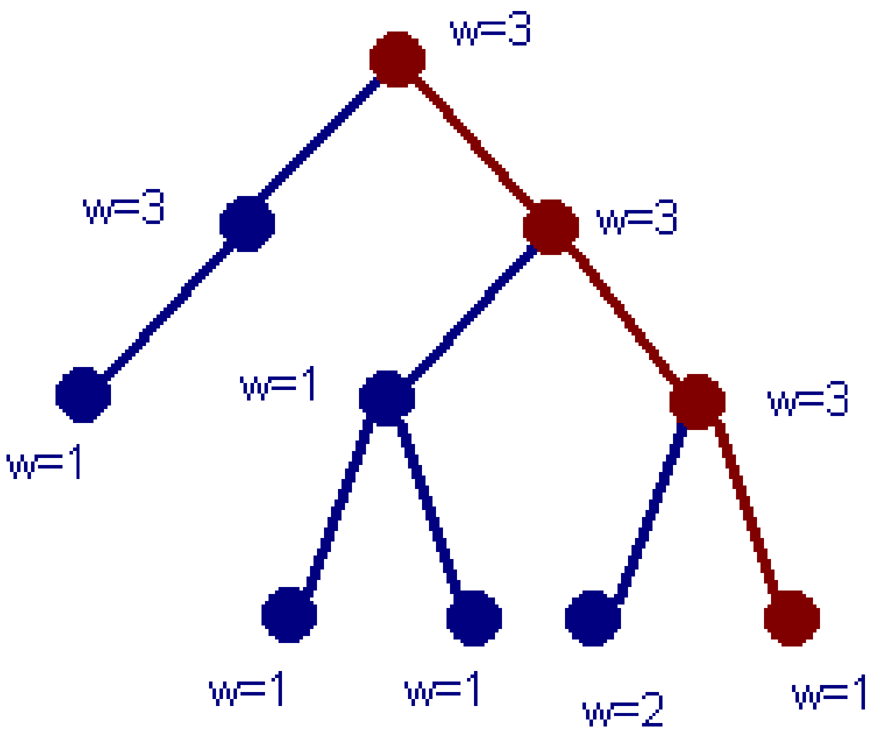
Third Path:
∑w = 8

On the example:



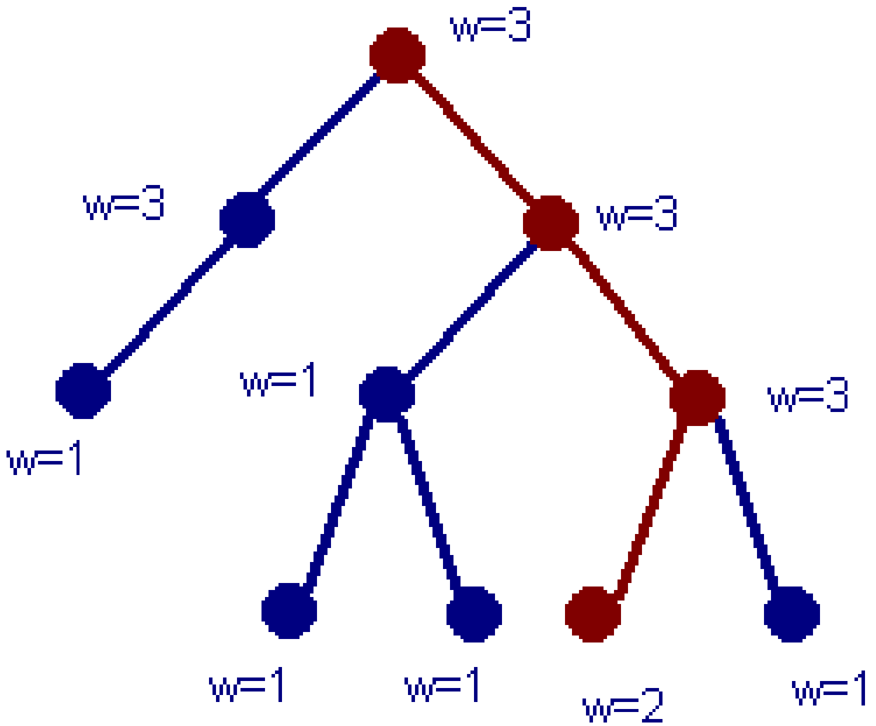
Fourth Path:
$\sum w = 11$

On the example:



Fifth Path:
∑w = 10

On the example:



PC1

How about the second PC?

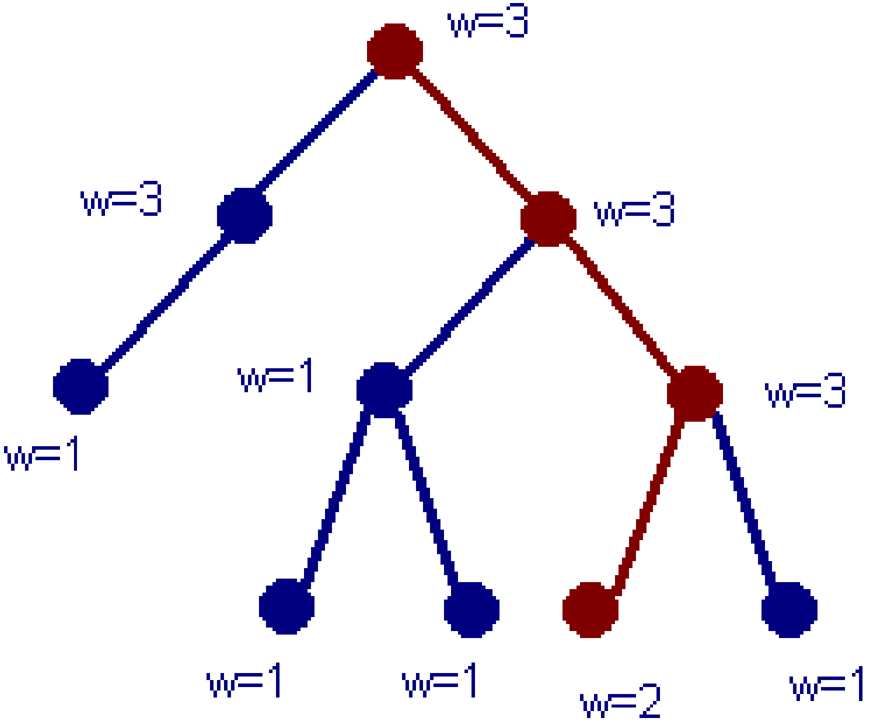
$$L_2^* = \arg\min_L \{ \sum_{t_i \in T} d(t_i, L_1 \cup L) \}$$

- Where we left the method was:

$$\sum_{t_i \in T} d(t_i, u_0) - \max_{L \in L_{u_0}} \sum_{v \in P(L)} w(v, T)$$

- Take the first PCA as the starting point, u_0

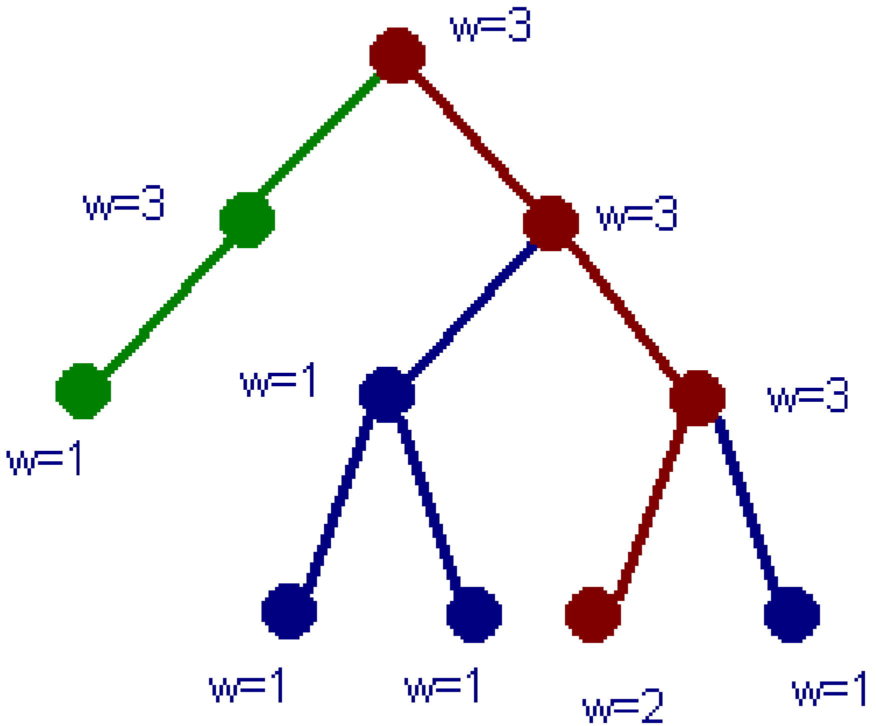
On the example:



PC1

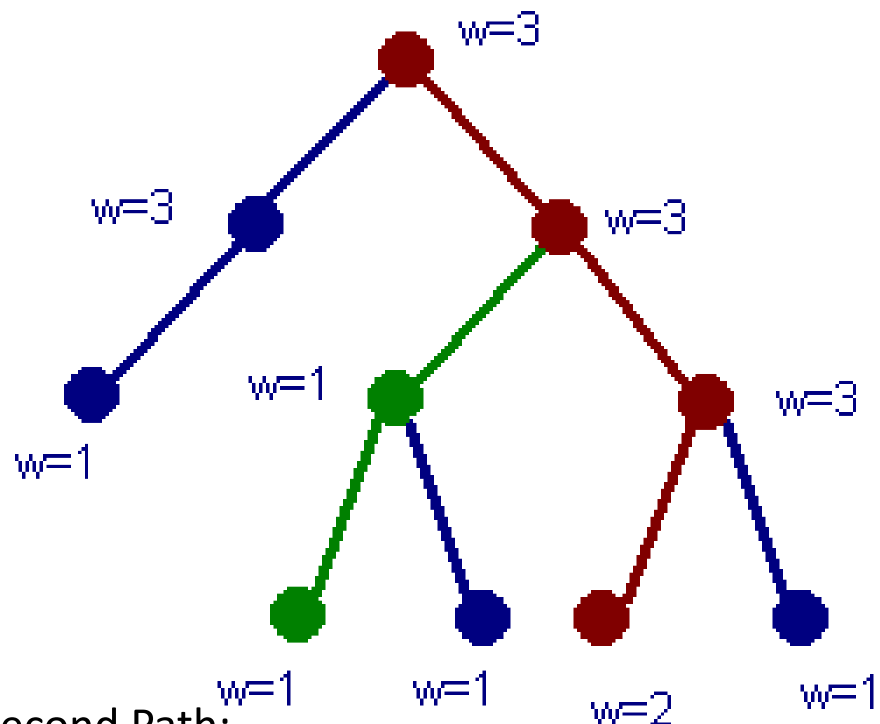
On the example:

First Path:
∑w = 4



PC1

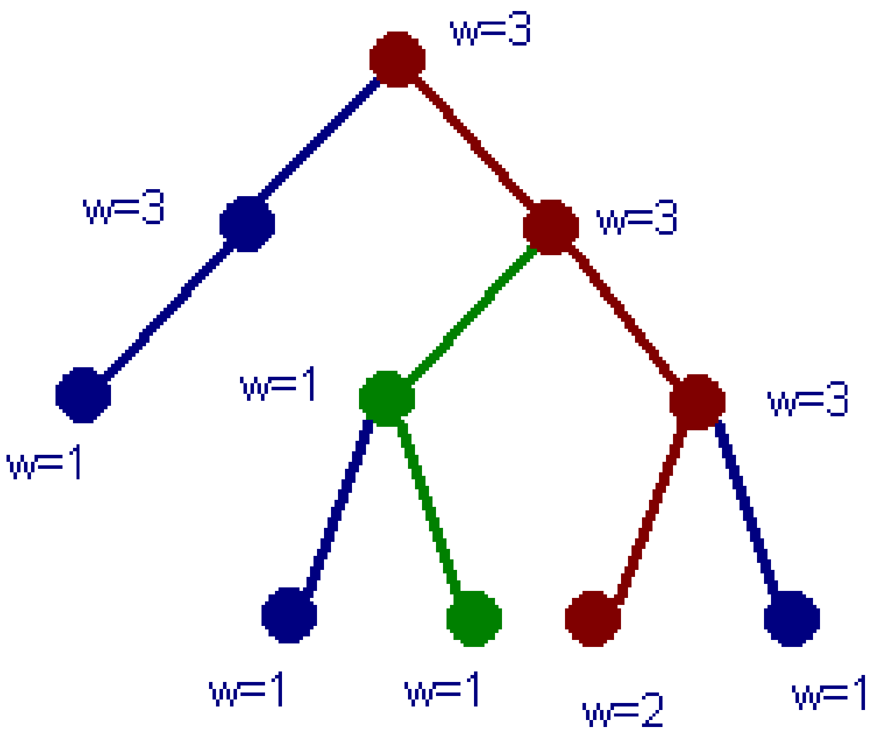
On the example:



Second Path:
$\sum w = 2$

PC1

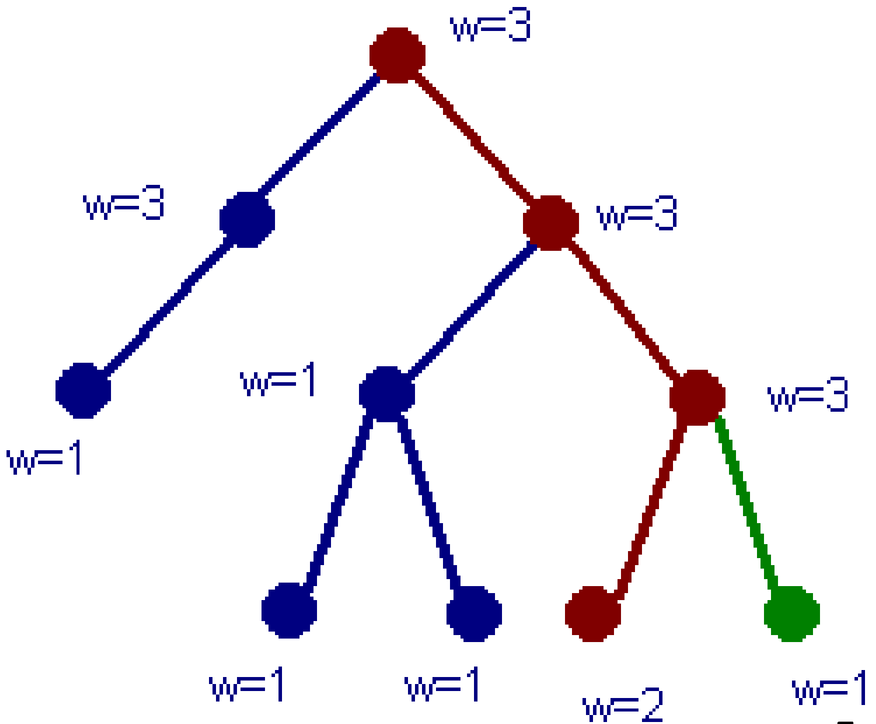
On the example:



Third Path:
$\sum w = 2$

PC1

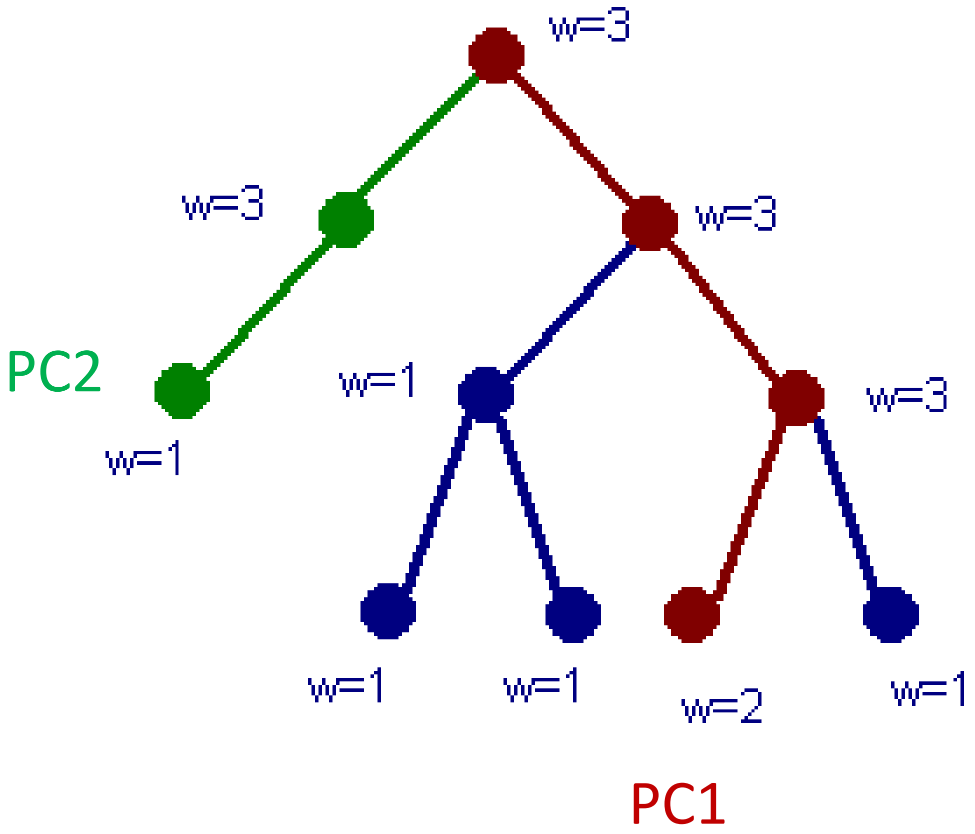
On the example:



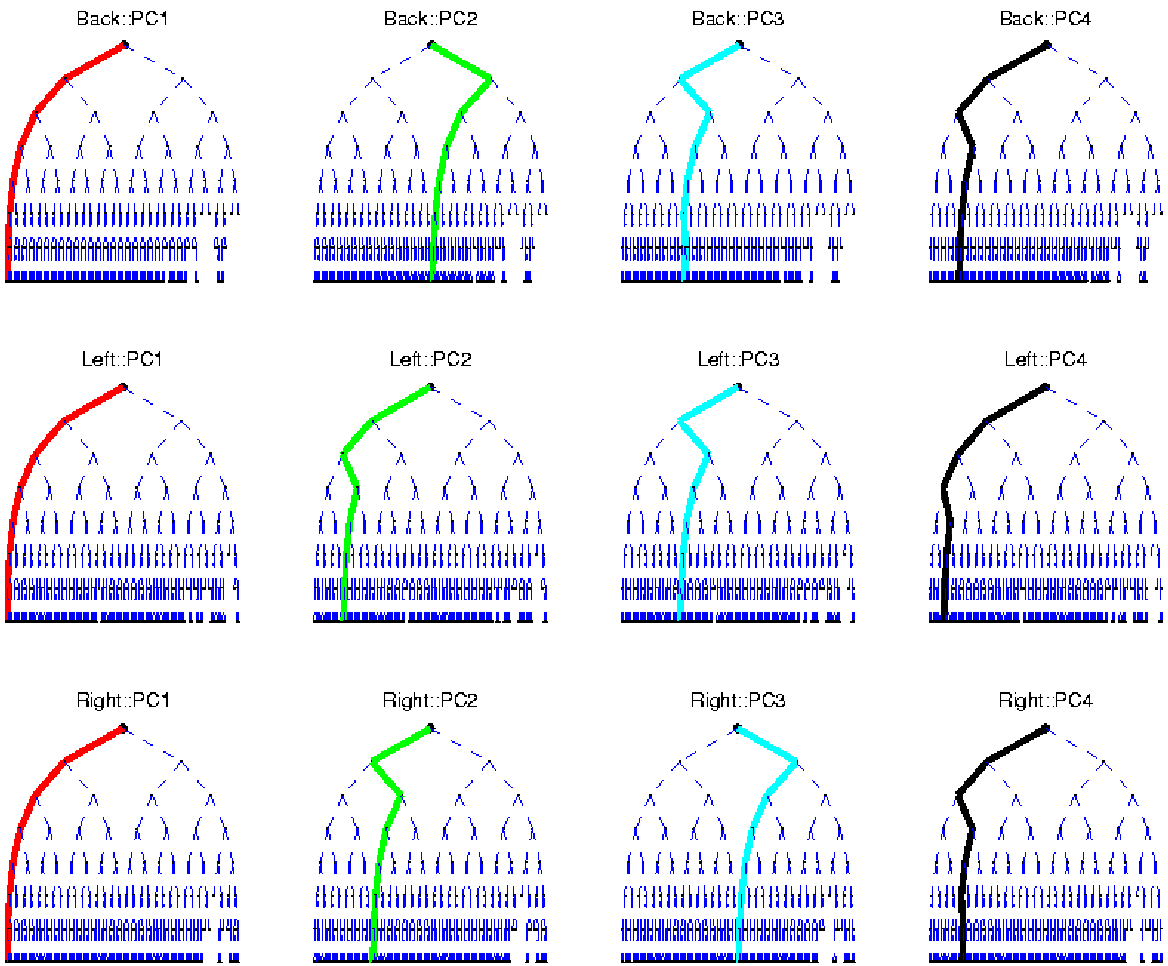
PC1

Fourth Path:
∑w = 1

On the example:



Real life examples:



An Extension

- Q: Are all nodes equally important?
- Assigning a weight, or, ‘importance’ to every node…
- Solved same way, only $w(v,T)$ is calculated differently.

Current research:

- TREECURVES
- Relax the requirement that every new node should be a child of the previous node

Some points

- Treecurve can go in and out of a tree
- Harder to compute
- Some heuristics are available
- No polynomial method that gives a guaranteed optimal is developed yet
- Q: Can the problem be NP-Hard?

Conclusion

- Treeline model, defined and solved
- Numerical analysis results explained by Prof. Marron
- New model, treecurves is the current research topic
- No optimal method yet

Questions, Comments?

Thank you!