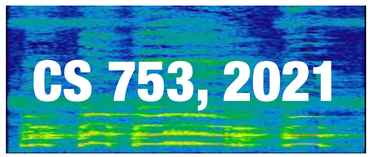
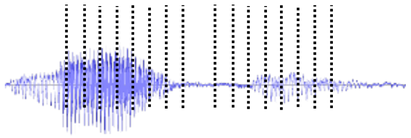
Tied-State HMMs for Acoustic Modeling

Lecture 2a

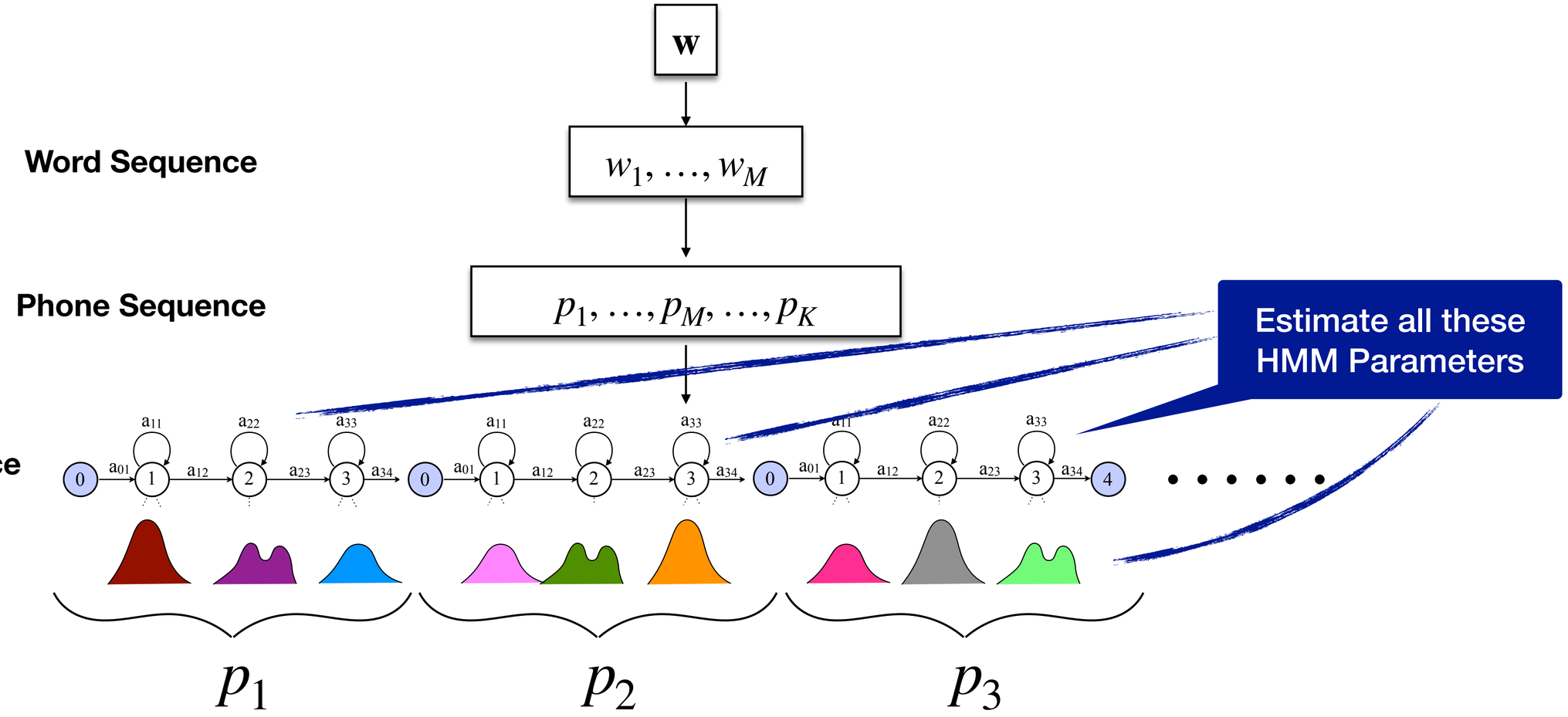


Instructor: Preethi Jyothi, IITB

Recap: HMM-based Acoustic Models

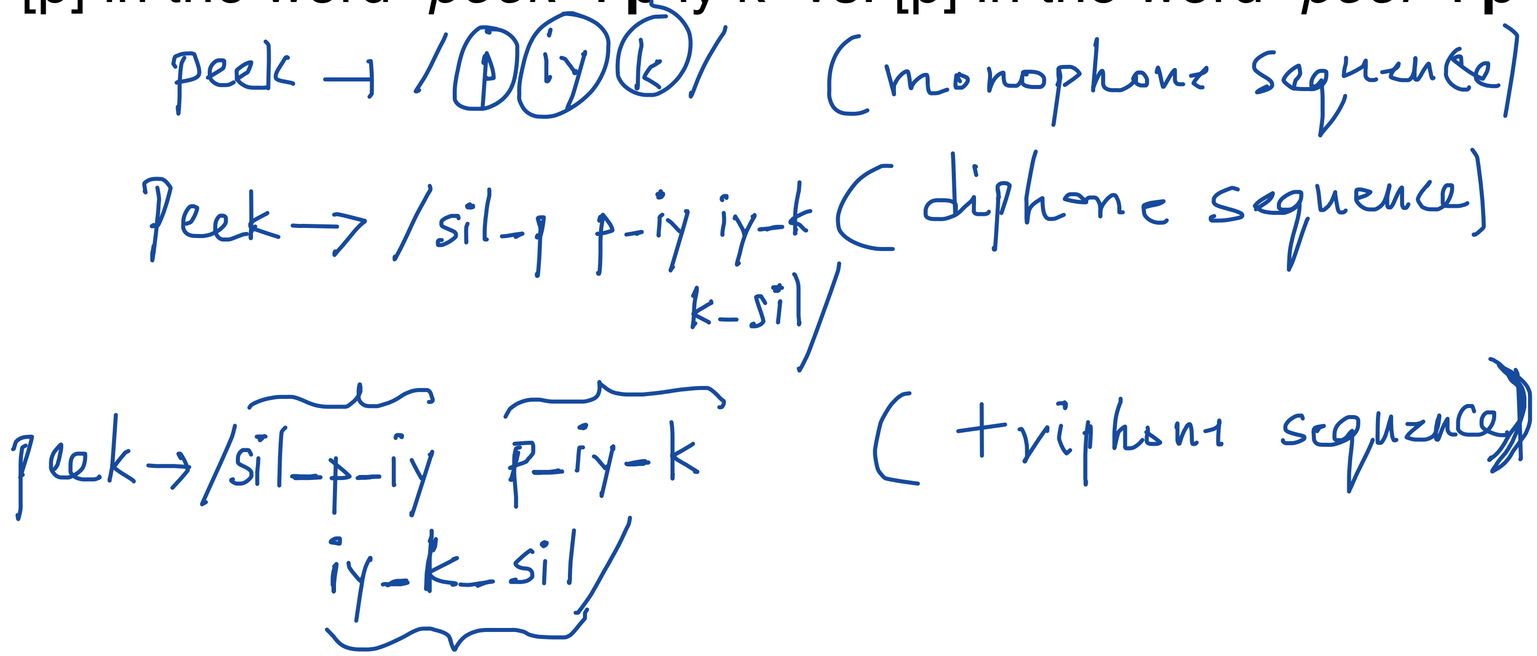


$\mathbf{O} = O_1, \ldots, O_T$ and $\mathbf{w} = w_1, \ldots, w_M$



Triphone HMM Models

- Each phone is modelled in the context of its left and right neighbour phones
  - Pronunciation of a phone is influenced by the preceding and succeeding phones.
    E.g. The phone [p] in the word "*peek*" : **p** iy k" vs. [p] in the word "*pool*" : **p** uw l



Triphone HMM Models

- Each phone is modelled in the context of its left and right neighbour phones
  - Pronunciation of a phone is influenced by the preceding and succeeding phones. E.g. The phone [p] in the word “*peek*” : **p** iy k” vs. [p] in the word “*pool*” : **p** uw l
- Number of triphones that appear in data ≈ 1000s or 10,000s

Triphone HMM Models

- Each phone is modelled in the context of its left and right neighbour phones
  - Pronunciation of a phone is influenced by the preceding and succeeding phones. E.g. The phone [p] in the word "*peek*" : **p** iy k" vs. [p] in the word "*pool*" : **p** uw l
- Number of triphones that appear in data ≈ 1000s or 10,000s
- If each triphone HMM has 3 states and each state generates m-component GMMs ($m \approx 64$), for d-dimensional acoustic feature vectors ($d \approx 40$) with Σ having d^2 parameters
  - Hundreds of millions of parameters!

Triphone HMM Models

- Each phone is modelled in the context of its left and right neighbour phones
  - Pronunciation of a phone is influenced by the preceding and succeeding phones. E.g. The phone [p] in the word "*peek*" : **p** iy k" vs. [p] in the word "*pool*" : **p** uw l
- Number of triphones that appear in data ≈ 1000s or 10,000s
- If each triphone HMM has 3 states and each state generates m-component GMMs ($m \approx 64$), for d-dimensional acoustic feature vectors ($d \approx 40$) with Σ having d^2 parameters
  - Hundreds of millions of parameters!
- Insufficient data to learn all triphone models reliably. What do we do? Share parameters across triphone models!

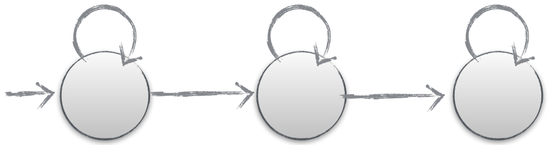
Parameter Sharing

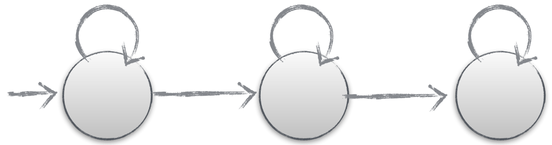
Parameter Sharing

- Sharing of parameters (also referred to as “parameter tying”) can be done at any level:
  - Parameters in HMMs corresponding to two triphones are said to be tied if they are identical

Parameter Sharing

- Sharing of parameters (also referred to as “parameter tying”) can be done at any level:
  - Parameters in HMMs corresponding to two triphones are said to be tied if they are identical



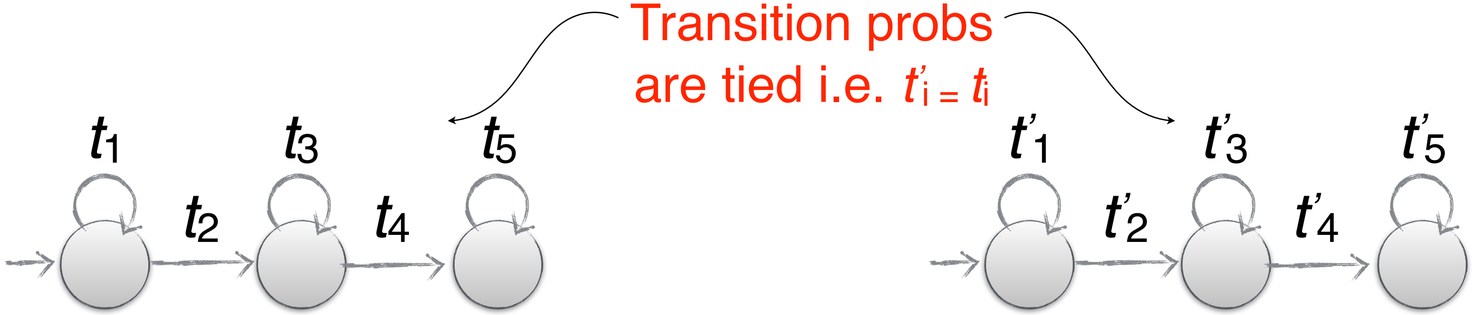


Parameter Sharing

- Sharing of parameters (also referred to as “parameter tying”) can be done at any level:
  - Parameters in HMMs corresponding to two triphones are said to be tied if they are identical

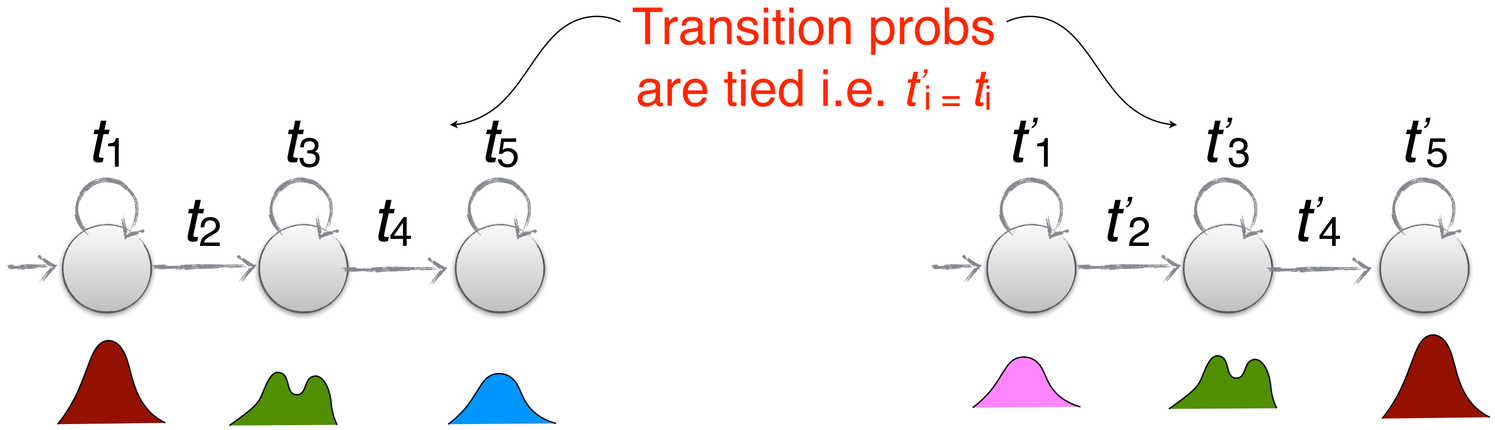
Parameter Sharing

- Sharing of parameters (also referred to as “parameter tying”) can be done at any level:
  - Parameters in HMMs corresponding to two triphones are said to be tied if they are identical



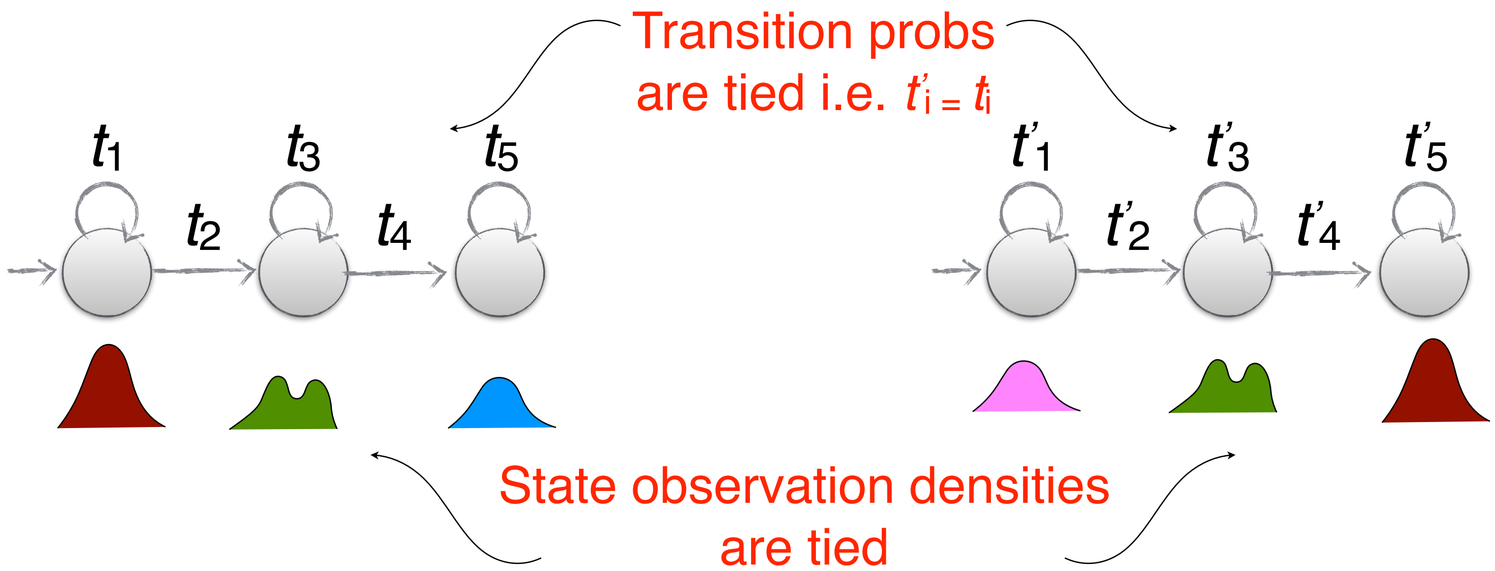
Parameter Sharing

- Sharing of parameters (also referred to as “parameter tying”) can be done at any level:
  - Parameters in HMMs corresponding to two triphones are said to be tied if they are identical



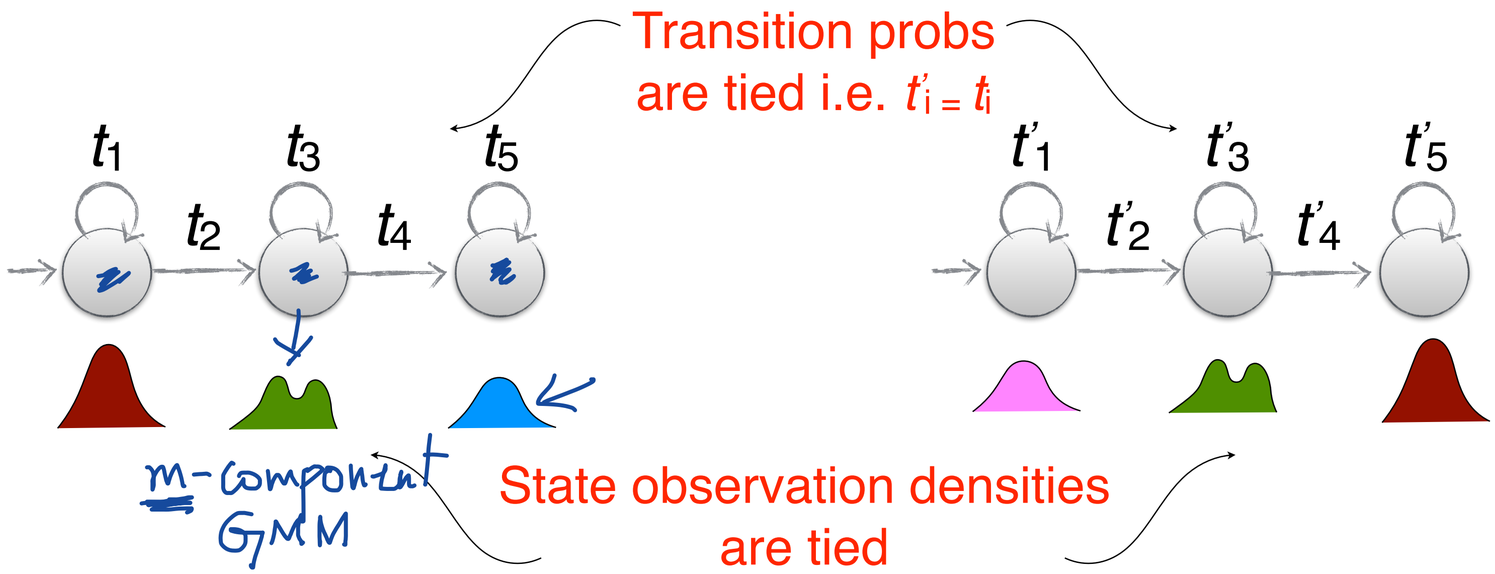
Parameter Sharing

- Sharing of parameters (also referred to as “parameter tying”) can be done at any level:
  - Parameters in HMMs corresponding to two triphones are said to be tied if they are identical



Parameter Sharing

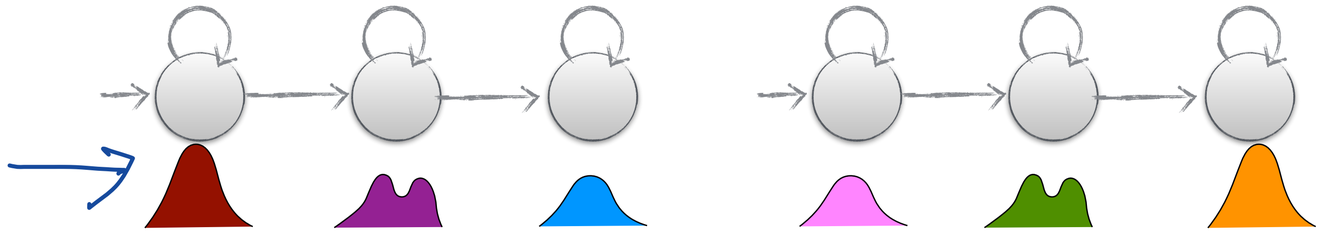
- Sharing of parameters (also referred to as “parameter tying”) can be done at any level:
  - Parameters in HMMs corresponding to two triphones are said to be tied if they are identical



- More parameter tying: Tying variances of all Gaussians within a state, tying variances of all Gaussians in all states, tying individual Gaussians, etc.

1. Tied Mixture Models

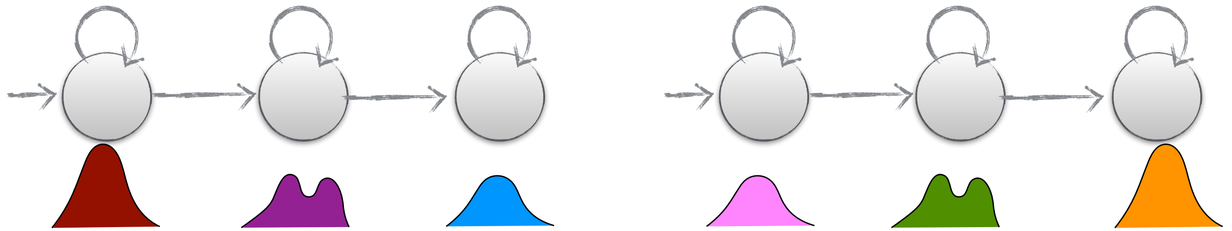
- All states share the same Gaussians (i.e. same means and covariances)
- Mixture weights are specific to each state



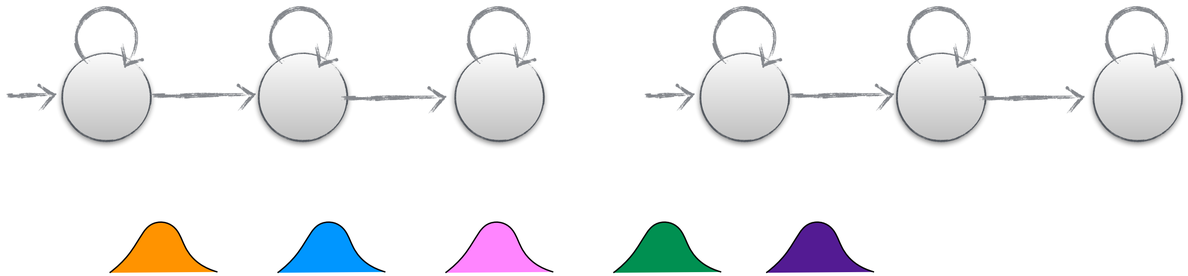
Triphone HMMs (No sharing)

1. Tied Mixture Models

- All states share the same Gaussians (i.e. same means and covariances)
- Mixture weights are specific to each state



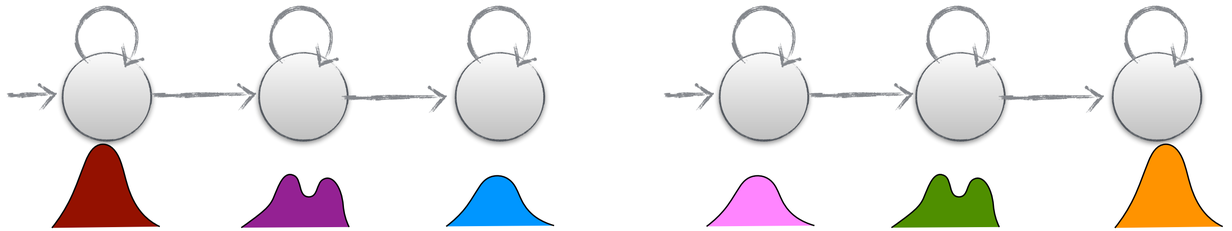
Triphone HMMs (No sharing)



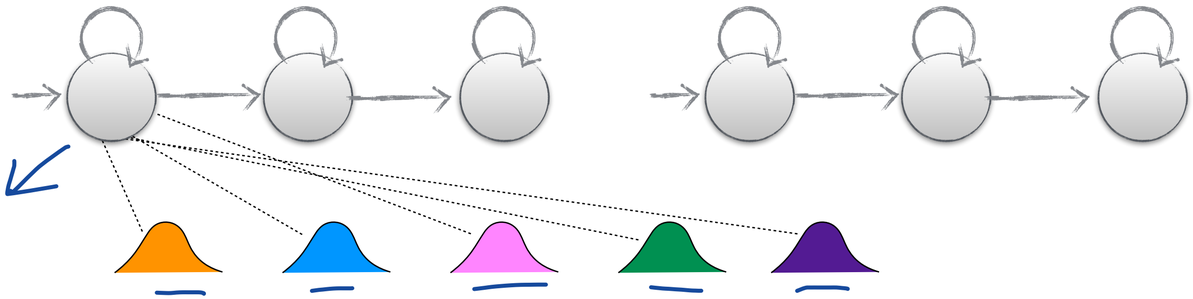
Triphone HMMs (Tied Mixture Models)

1. Tied Mixture Models

- All states share the same Gaussians (i.e. same means and covariances)
- Mixture weights are specific to each state



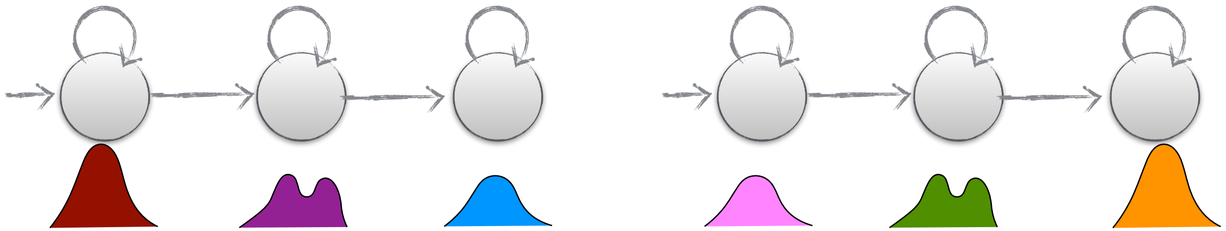
Triphone HMMs (No sharing)



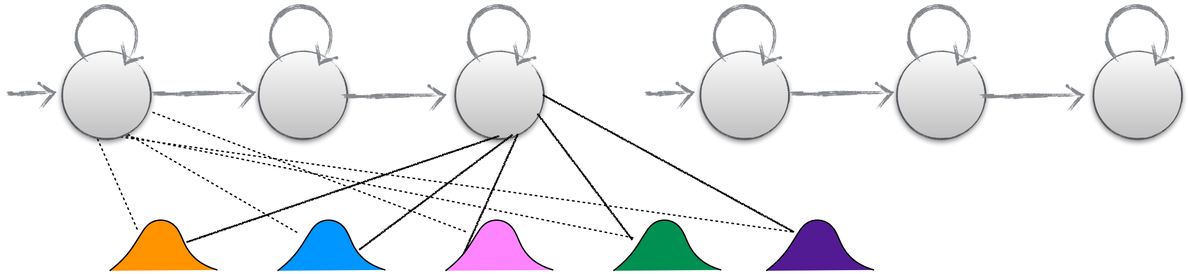
Triphone HMMs (Tied Mixture Models)

1. Tied Mixture Models

- All states share the same Gaussians (i.e. same means and covariances)
- Mixture weights are specific to each state



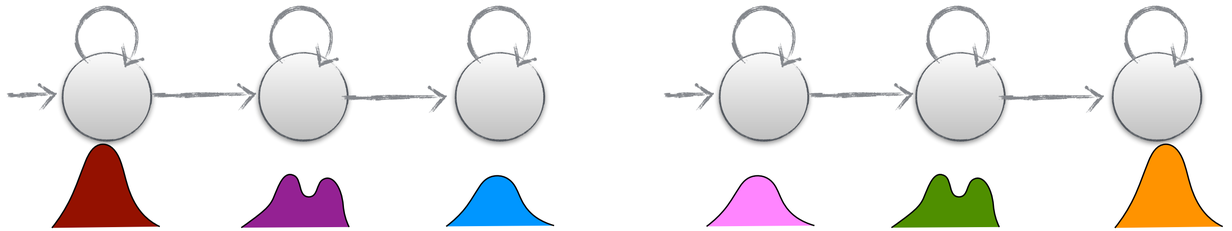
Triphone HMMs (No sharing)



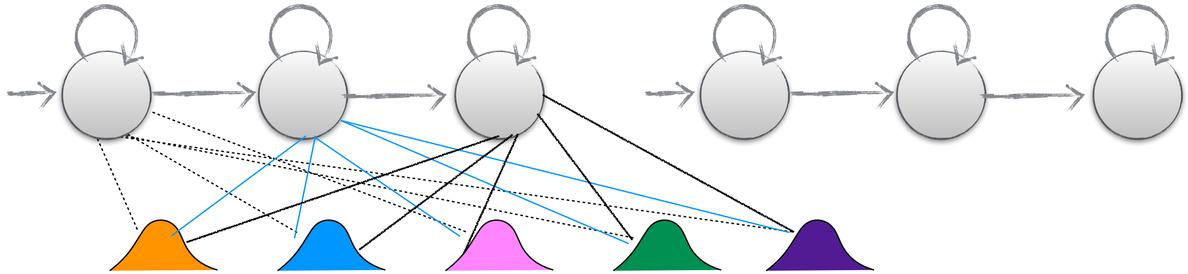
Triphone HMMs (Tied Mixture Models)

1. Tied Mixture Models

- All states share the same Gaussians (i.e. same means and covariances)
- Mixture weights are specific to each state



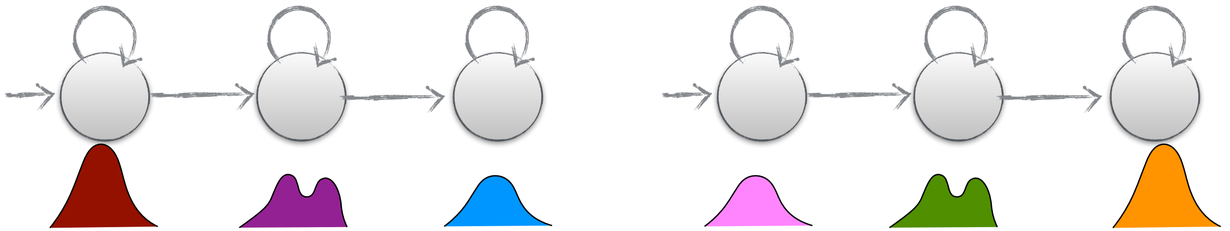
Triphone HMMs (No sharing)



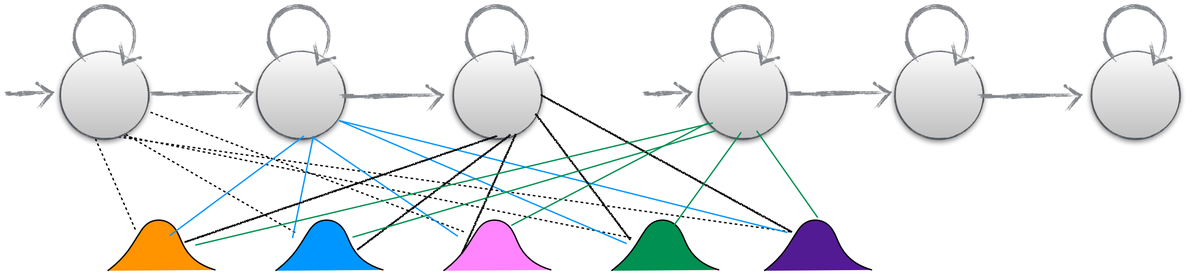
Triphone HMMs (Tied Mixture Models)

1. Tied Mixture Models

- All states share the same Gaussians (i.e. same means and covariances)
- Mixture weights are specific to each state



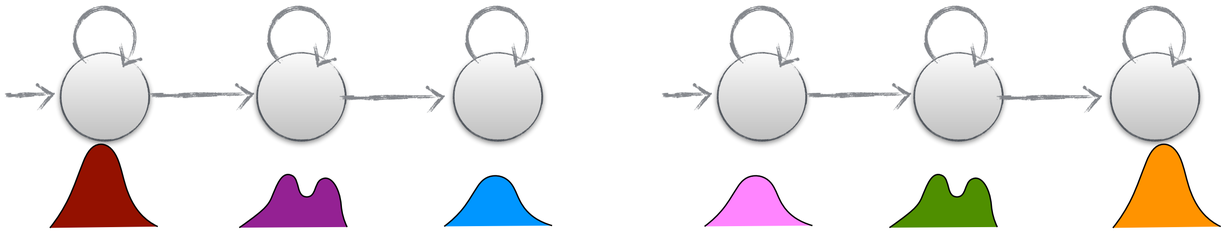
Triphone HMMs (No sharing)



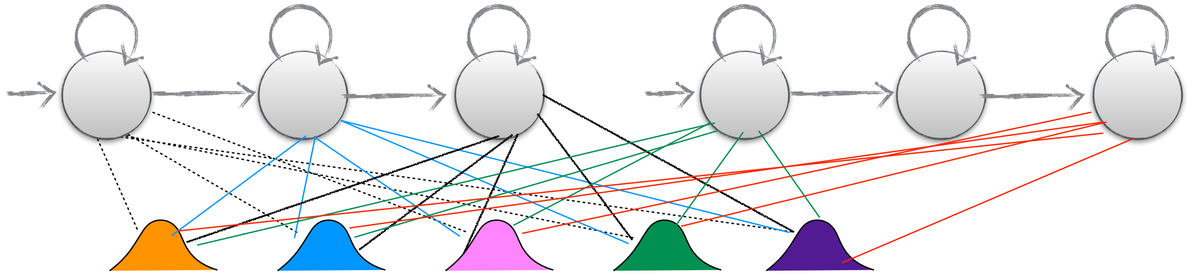
Triphone HMMs (Tied Mixture Models)

1. Tied Mixture Models

- All states share the same Gaussians (i.e. same means and covariances)
- Mixture weights are specific to each state



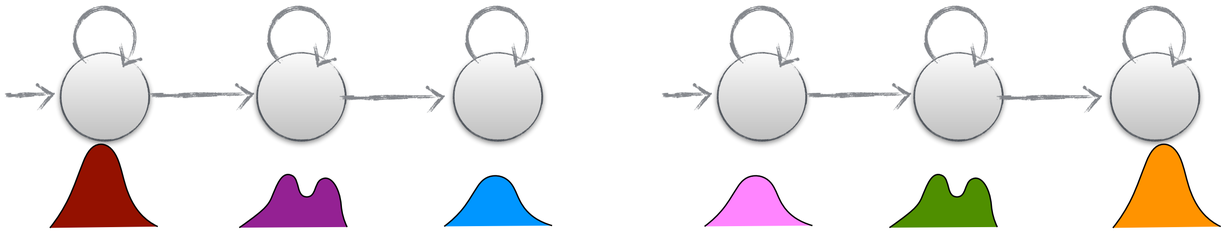
Triphone HMMs (No sharing)



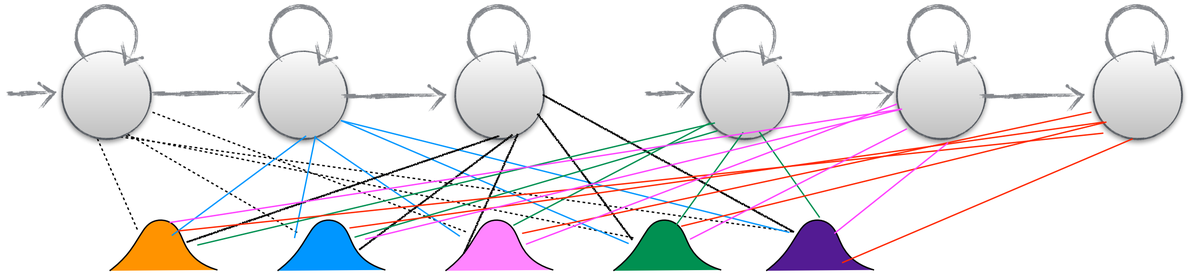
Triphone HMMs (Tied Mixture Models)

1. Tied Mixture Models

- All states share the same Gaussians (i.e. same means and covariances)
- Mixture weights are specific to each state



Triphone HMMs (No sharing)



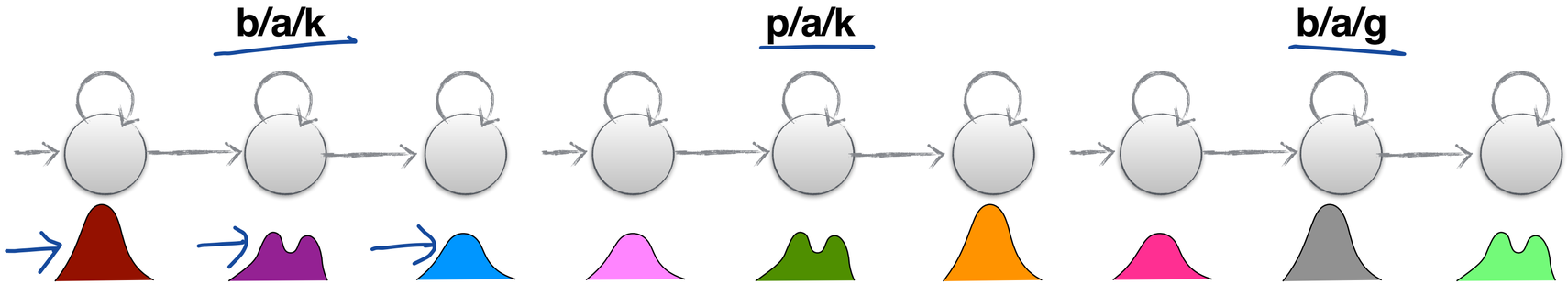
Triphone HMMs (Tied Mixture Models)

2. State Tying

- Observation probabilities are shared across states which generate acoustically similar data

2. State Tying

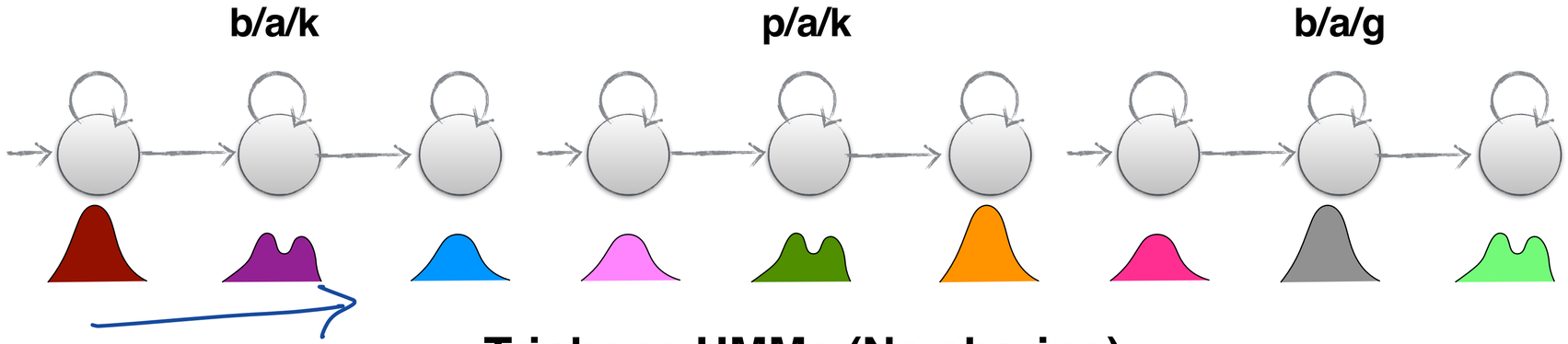
- Observation probabilities are shared across states which generate acoustically similar data



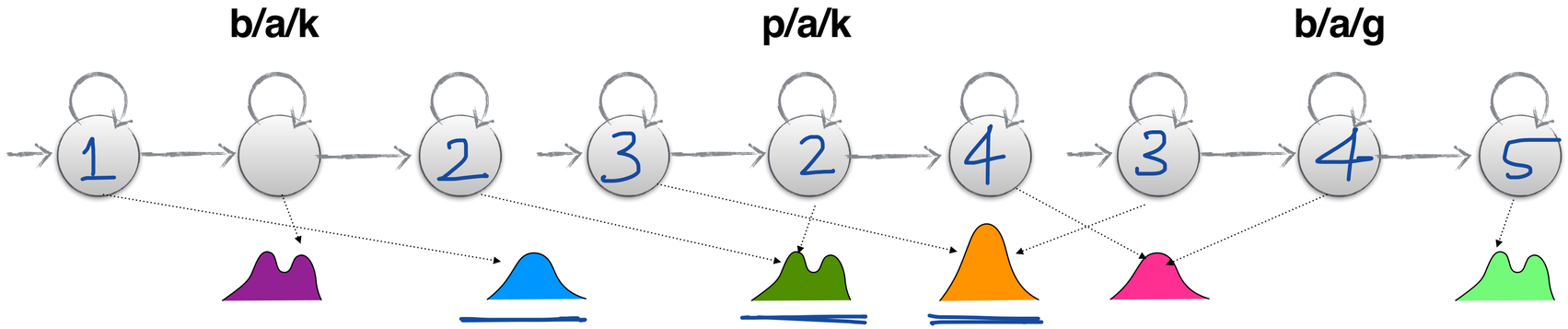
Triphone HMMs (No sharing)

2. State Tying

- Observation probabilities are shared across states which generate acoustically similar data



Triphone HMMs (No sharing)



Triphone HMMs (State Tying)

Tied state HMMs: Step 1

Create and train 3-state monophone HMMs with single Gaussian observation probability densities

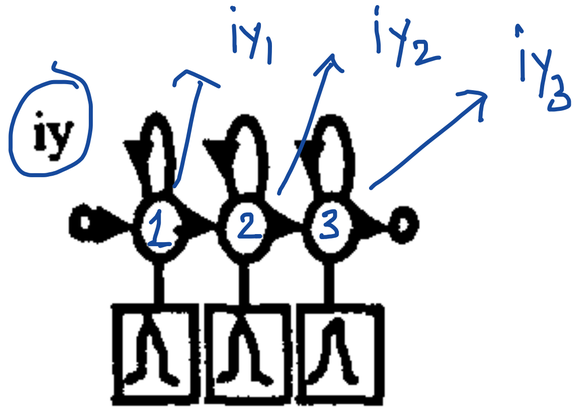


Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs: Step 2

Clone these monophone distributions to initialise a set of untied triphone models

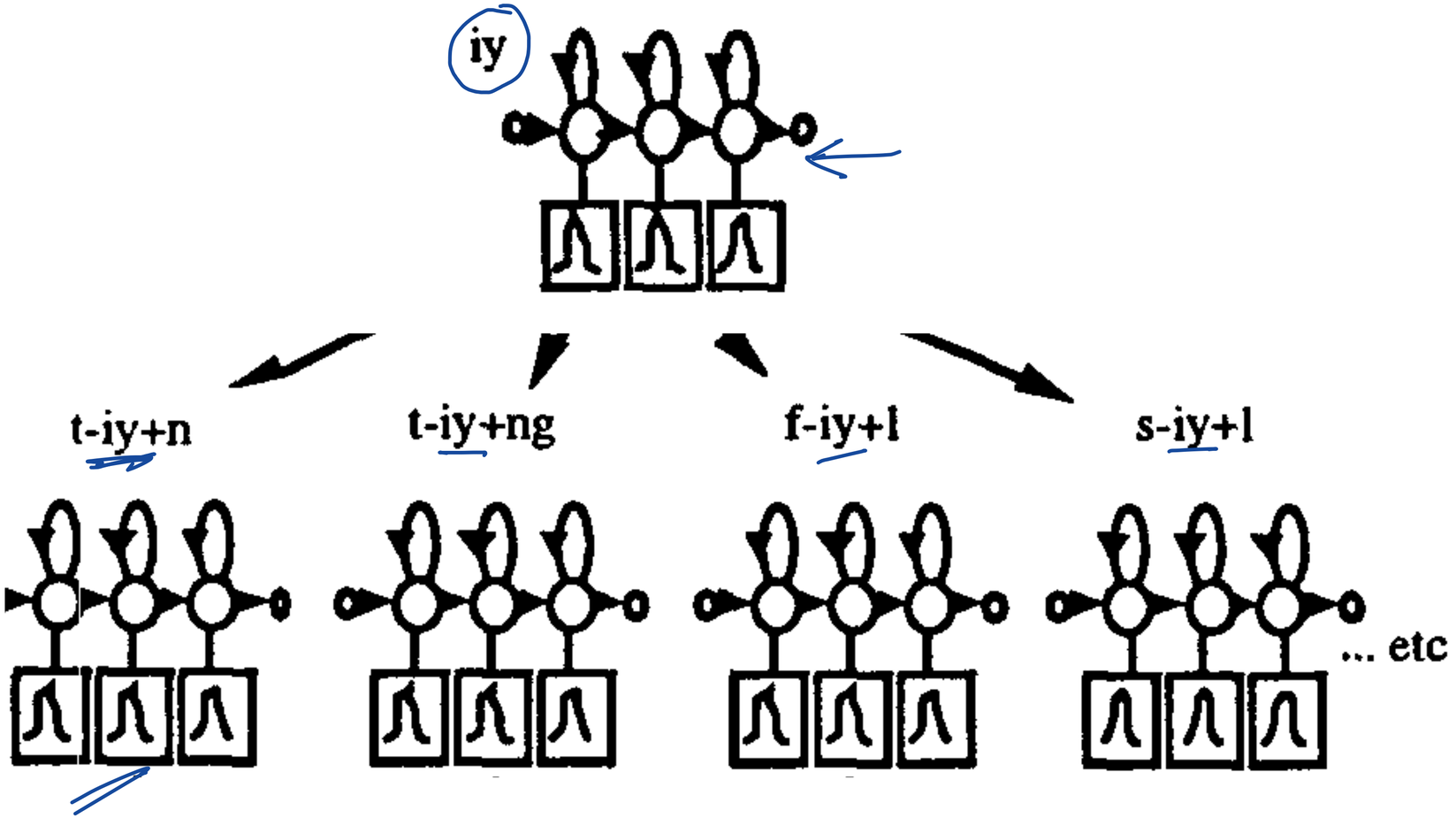
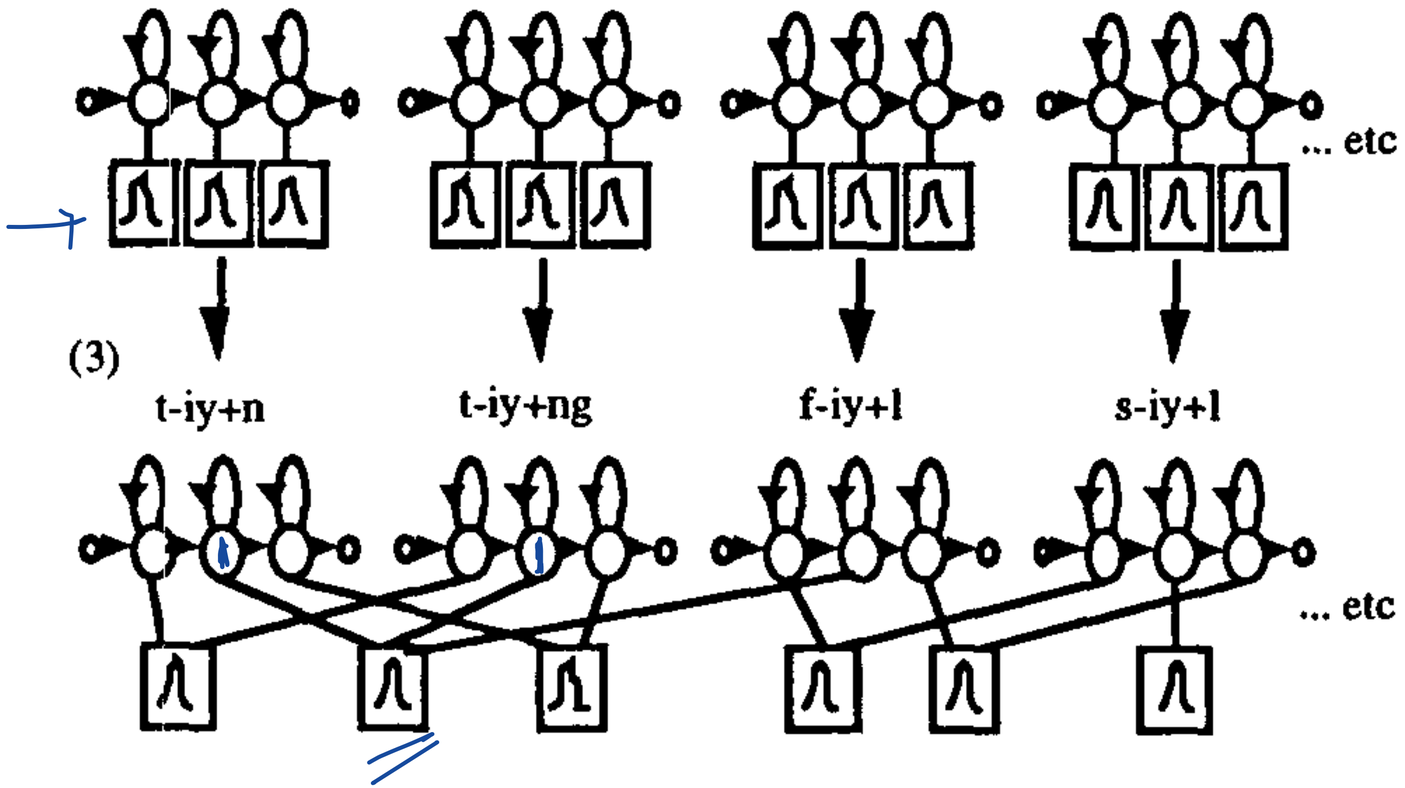


Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

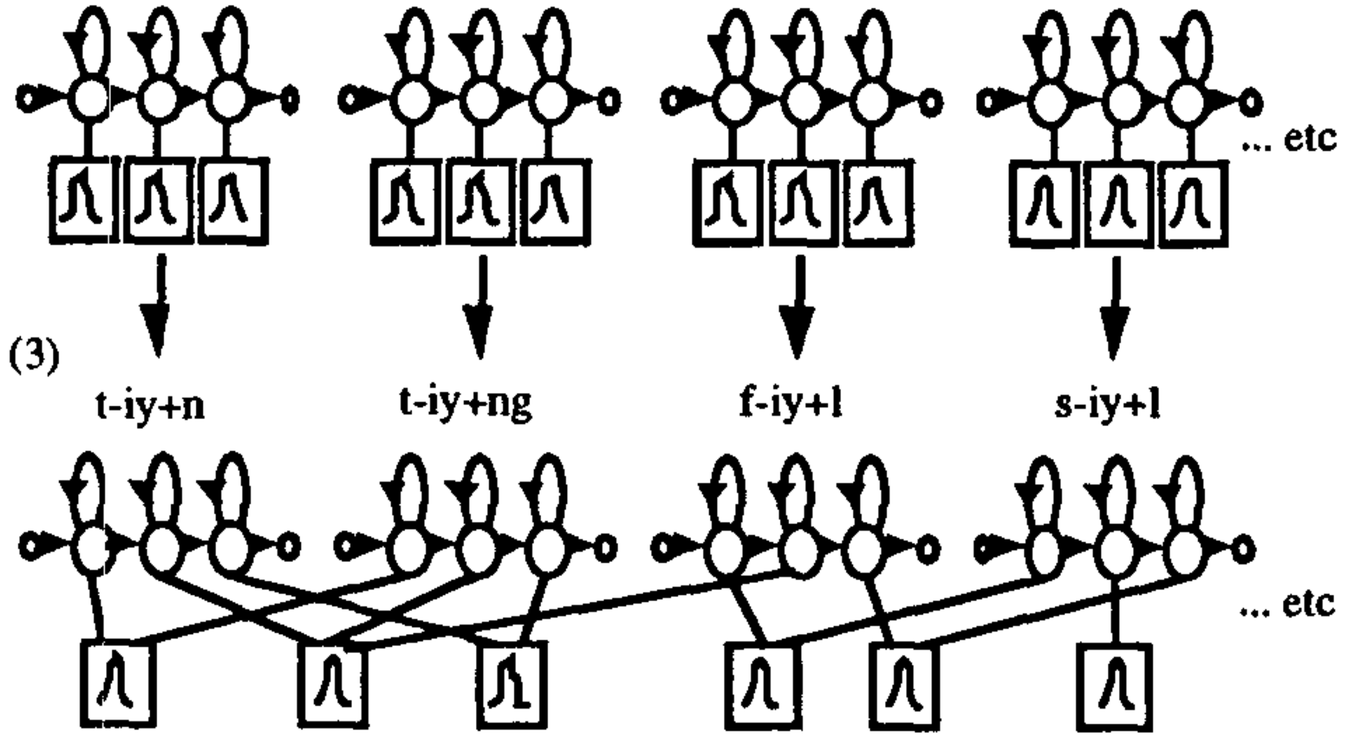
Tied state HMMs: Step 3



For all triphones derived from the same monophone, cluster states whose parameters should be tied together.

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs: Step 3



For all triphones derived from the same monophone, cluster states whose parameters should be tied together.

Popular option: Use ***decision trees*** to determine which states should be tied together!

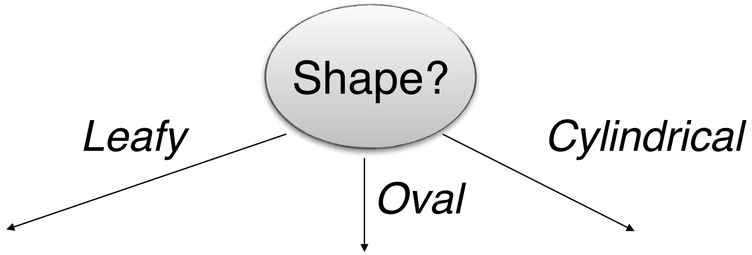
Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Decision Trees

Decision Trees

Classification using a decision tree:

Begins at the root node: What property is satisfied?

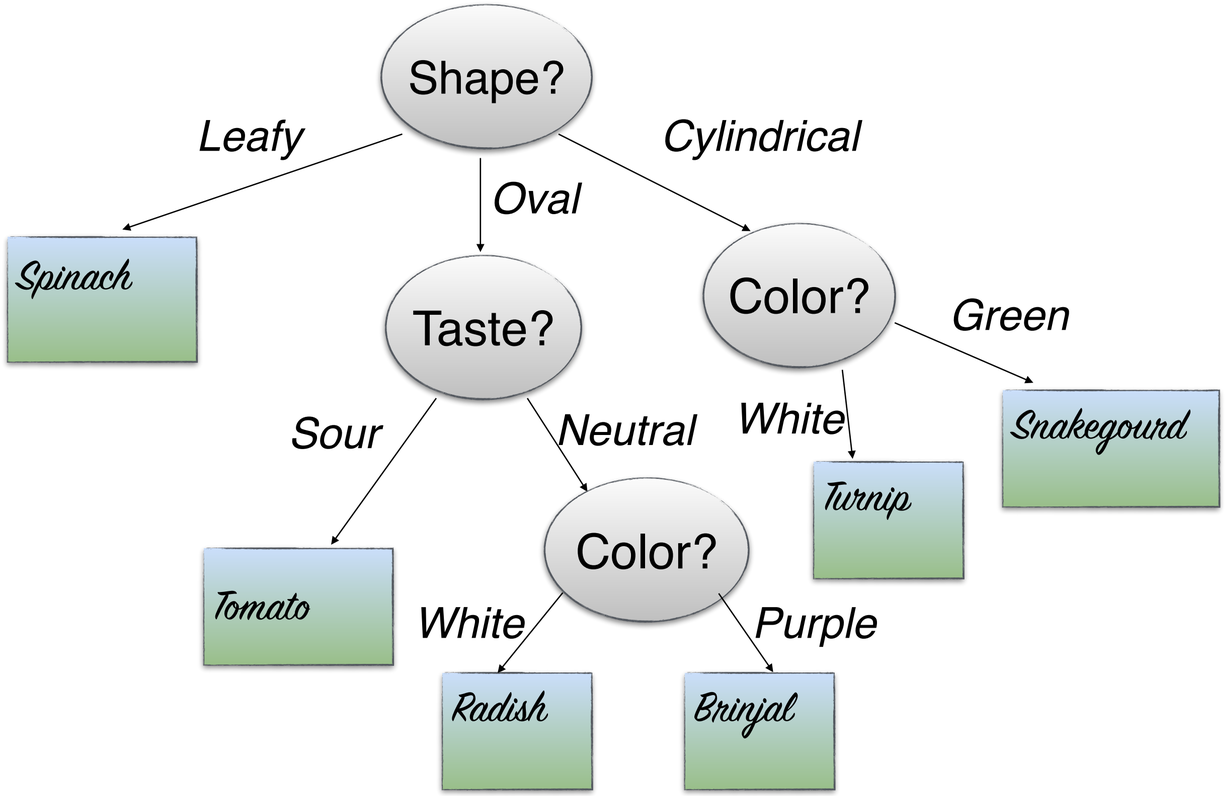


Decision Trees

Classification using a decision tree:

Begins at the root node: What property is satisfied?

Depending on answer, traverse to different branches



Decision Trees

Decision Trees

- Given the data at a node, either declare the node to be a leaf or find another attribute to split the node further.

Decision Trees

- Given the data at a node, either declare the node to be a leaf or find another attribute to split the node further.

- Important questions to be addressed for DTs:

Decision Trees

- Given the data at a node, either declare the node to be a leaf or find another attribute to split the node further.

- Important questions to be addressed for DTs:

  1. How many splits at a node?
     Chosen by the user.

Decision Trees

- Given the data at a node, either declare the node to be a leaf or find another attribute to split the node further.

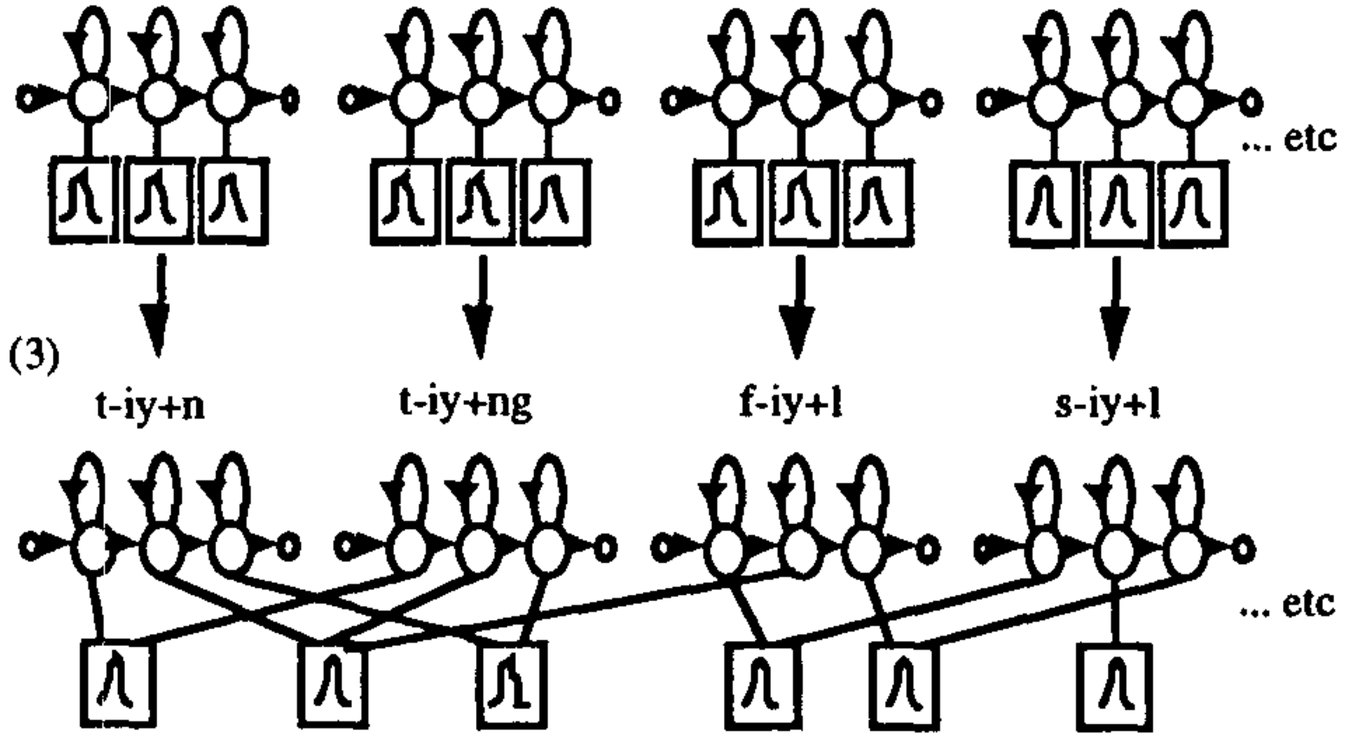
- Important questions to be addressed for DTs:
  1. How many splits at a node?
     Chosen by the user.
  2. Which attribute/question should be used at a node for splitting?
     One which decreases “impurity” of nodes as much as possible.

Decision Trees

- Given the data at a node, either declare the node to be a leaf or find another attribute to split the node further.

- Important questions to be addressed for DTs:
  1. How many splits at a node?
     Chosen by the user.
  2. Which attribute/question should be used at a node for splitting?
     One which decreases “impurity” of nodes as much as possible.
  3. When is a node a leaf?
     Set threshold in reduction in impurity

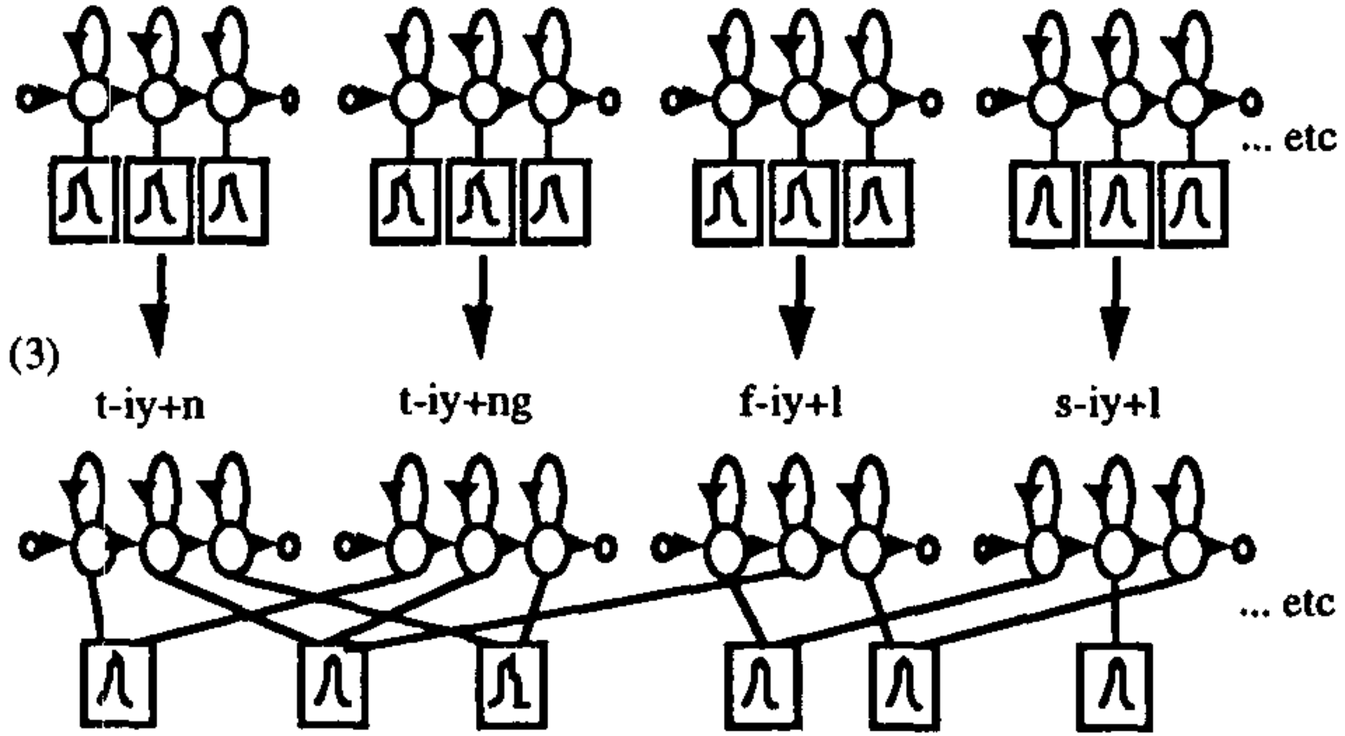
Tied state HMMs: Step 3



For all triphones derived from the same monophone, cluster states whose parameters should be tied together.

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs: Step 3



For all triphones derived from the same monophone, cluster states whose parameters should be tied together.

Popular option: Use ***decision trees*** to determine which states should be tied together!

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Example: Phonetic Decision Tree (DT)

One tree is constructed for each state of each monophone to cluster all the corresponding triphone states

OW1 OW2 OW3

Head node
aa2/ow2/f2, aa2/ow2/s2,
aa2/ow2/d2, h2/ow2/p2,
aa2/ow2/n2, aa2/ow2/g2,
…

DT for center state of [ow]

Uses all training data tagged with *-*ow*$_2$+*

$O_1, \ldots, O_T$

$P_1, \ldots, P_K$

$O_1, \ldots, O_T$

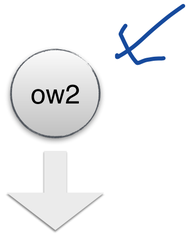
$P'_1, \ldots, P'_T$

$P'_1 = P_1$

$P'_T = P_K$

Example: Phonetic Decision Tree (DT)

One tree is constructed for each state of each monophone to cluster all the corresponding triphone states



Head node
aa_2/ow_2/f_2, aa_2/ow_2/s_2,
aa_2/ow_2/d_2, h_2/ow_2/p_2,
aa_2/ow_2/n_2, aa_2/ow_2/g_2,
…

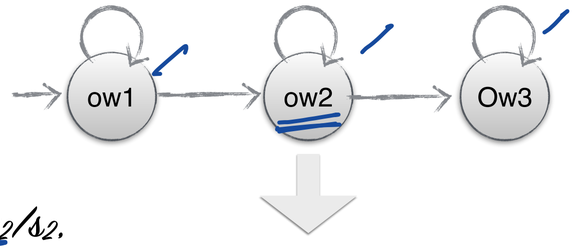
DT for center state of [ow]

Uses all training data tagged with *-ow_2+*

How do we determine this training data? Coming up in two slides.

Example: Phonetic Decision Tree (DT)

One tree is constructed for each state of each monophone to cluster all the corresponding triphone states



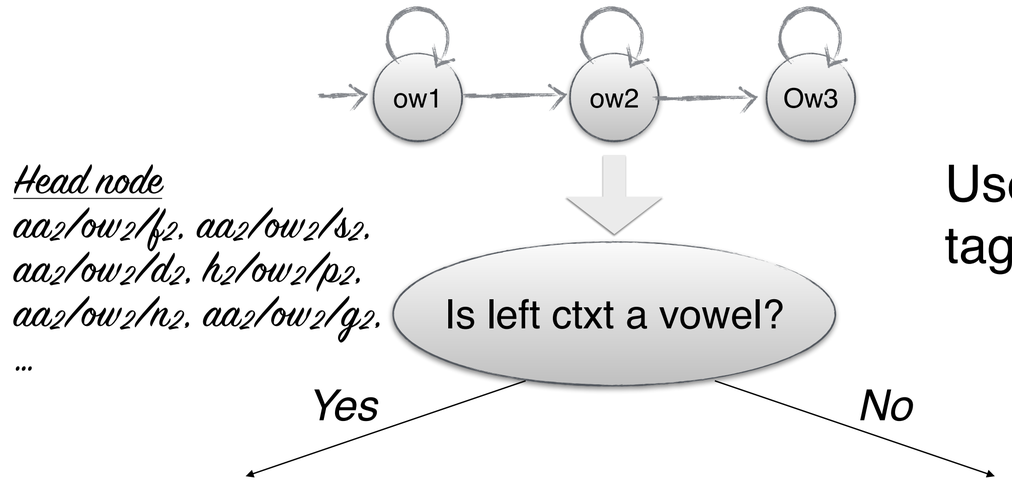
Head node
$aa_2/ow_2/f_2$, $aa_2/ow_2/s_2$,
$aa_2/ow_2/d_2$, $h_2/ow_2/p_2$,
$aa_2/ow_2/n_2$, $aa_2/ow_2/g_2$,
…

DT for center state of [ow]

Uses all training data tagged as *-ow_2+*

Example: Phonetic Decision Tree (DT)

One tree is constructed for each state of each monophone to cluster all the corresponding triphone states

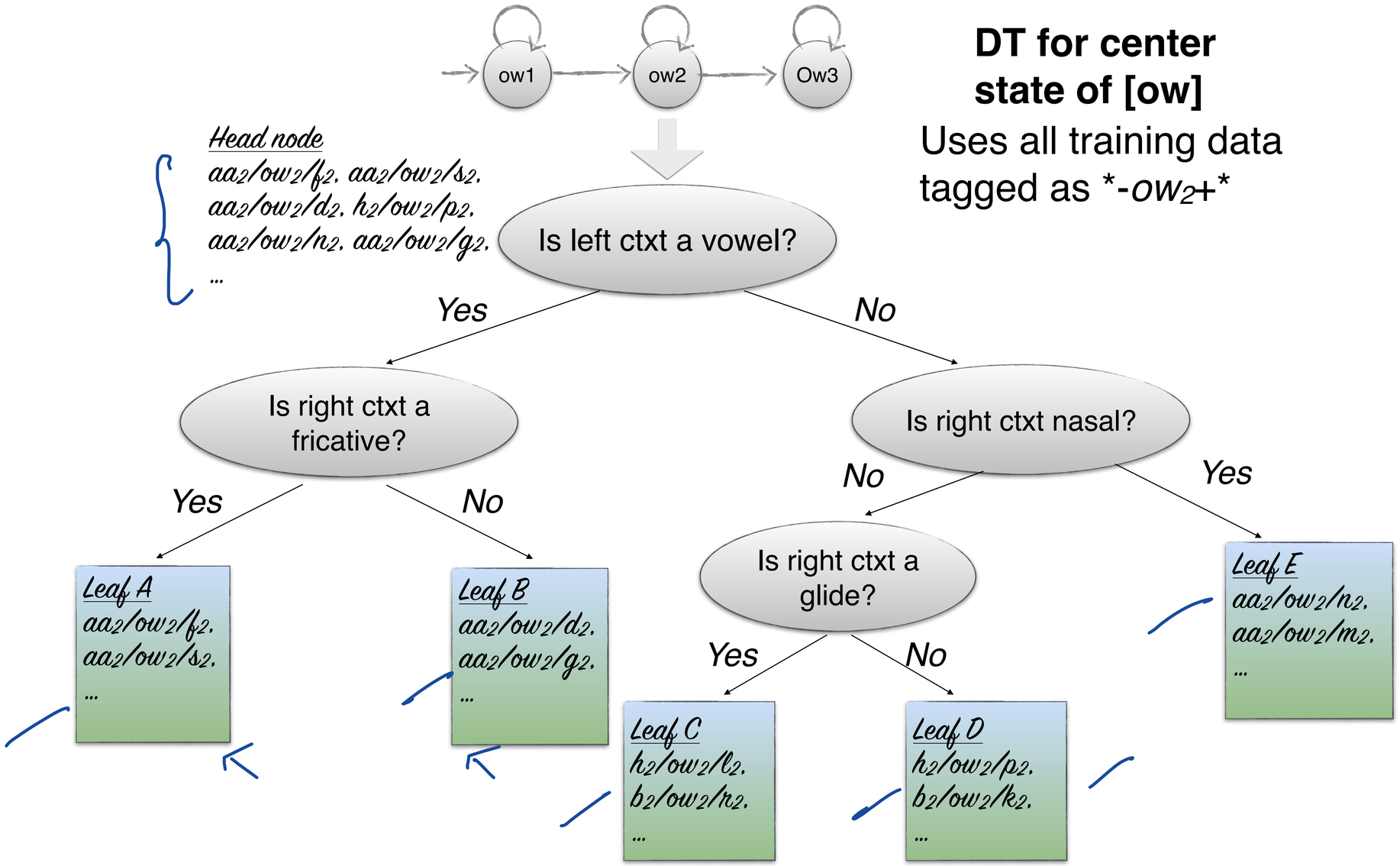


DT for center state of [ow]

Uses all training data tagged as *-ow_2+*

Example: Phonetic Decision Tree (DT)

One tree is constructed for each state of each monophone to cluster all the corresponding triphone states

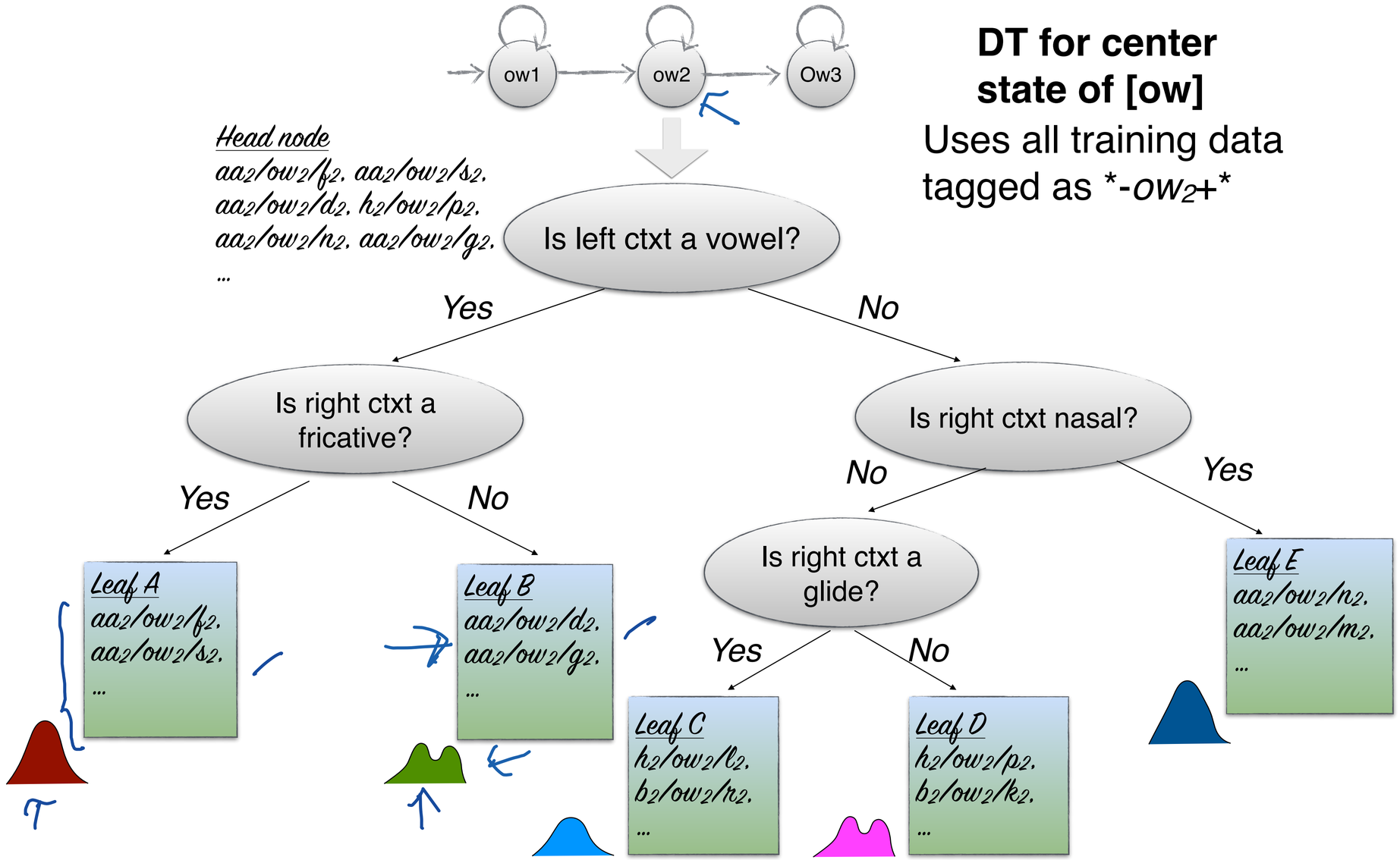


DT for center state of [ow]

Uses all training data tagged as *-ow_2+*

Example: Phonetic Decision Tree (DT)

One tree is constructed for each state of each monophone to cluster all the corresponding triphone states



DT for center state of [ow]

Uses all training data tagged as *-ow_2+*

How do we build these phone DTs?

1. What questions are used?

   Linguistically-inspired binary questions: “Does the left or right phone come from a broad class of phones such as vowels, stops, etc.?” “Is the left or right phone [k] or [m]?”

How do we build these phone DTs?

1. What questions are used?

   Linguistically-inspired binary questions: “Does the left or right phone come from a broad class of phones such as vowels, stops, etc.?” “Is the left or right phone [k] or [m]?”

   ow$_2$

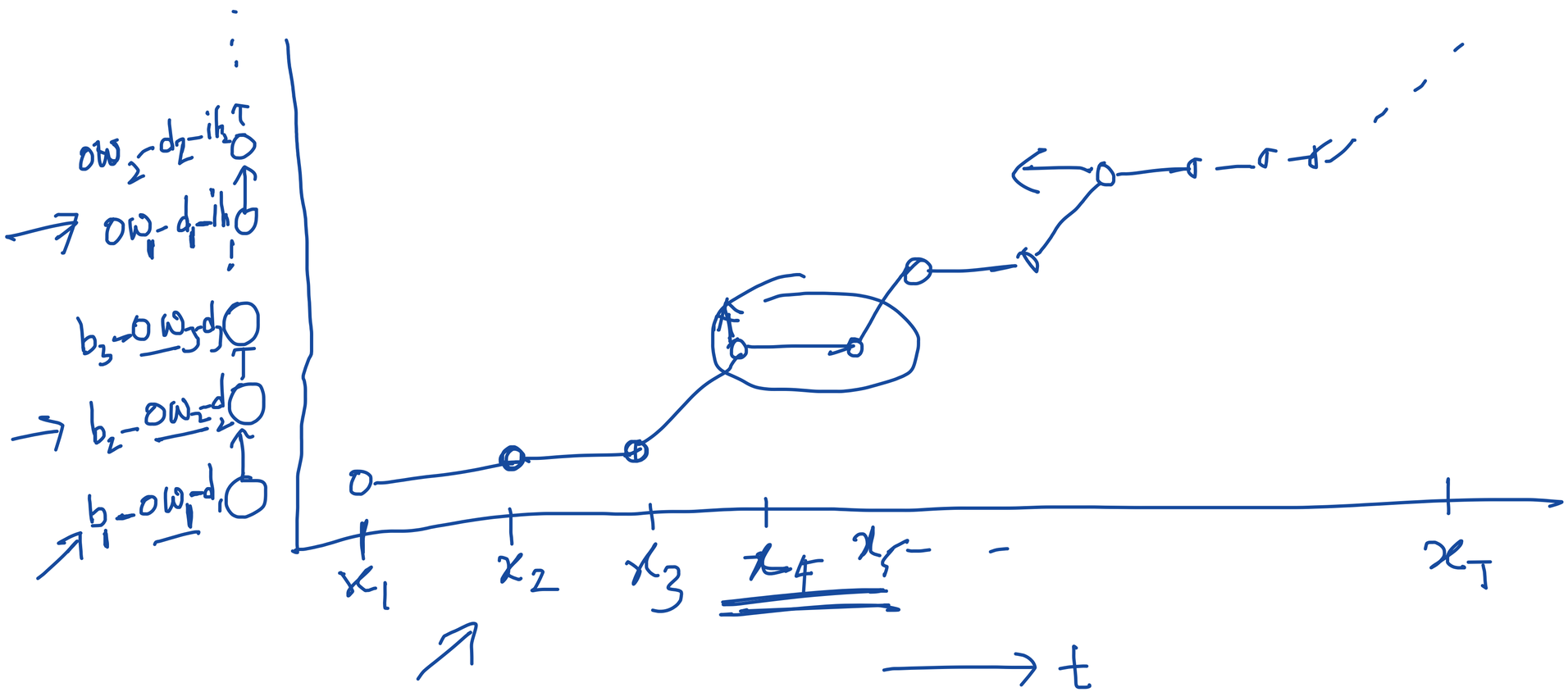
2. What is the training data for each phone state, p_j? (root node of DT)

   All speech frames that align with the j^{th} state of every triphone HMM that has p as the middle phone

Training data for DT nodes

Training data for DT nodes

- Align training instance $x = (x_1, \ldots, x_T)$ where $x_i \in \mathbb{R}^d$ with a set of triphone HMMs
- Use Viterbi algorithm to find the best HMM triphone state sequence corresponding to each x

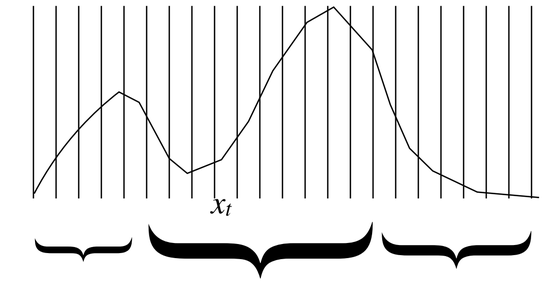


Training data for DT nodes

- Align training instance $x = (x_1, \ldots, x_T)$ where $x_i \in \mathbb{R}^d$ with a set of triphone HMMs
- Use Viterbi algorithm to find the best HMM triphone state sequence corresponding to each x
- Tag each x_t with ID of current phone along with left-context and right-context

Training data for DT nodes

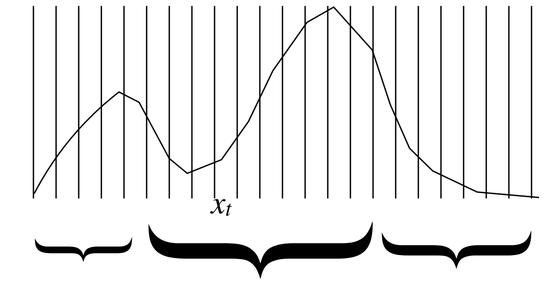
- Align training instance $x = (x_1, \ldots, x_T)$ where $x_i \in \mathbb{R}^d$ with a set of triphone HMMs
- Use Viterbi algorithm to find the best HMM triphone state sequence corresponding to each x
- Tag each x_t with ID of current phone along with left-context and right-context



x_t is tagged with ID b_2-aa_2+g_2 i.e. x_t is aligned with the second state of the 3-state HMM corresponding to the triphone b-aa+g

Training data for DT nodes

- Align training instance $x = (x_1, \ldots, x_T)$ where $x_i \in \mathbb{R}^d$ with a set of triphone HMMs
- Use Viterbi algorithm to find the best HMM triphone state sequence corresponding to each x
- Tag each x_t with ID of current phone along with left-context and right-context



x_t is tagged with ID b_2-aa_2+g_2 i.e. x_t is aligned with the second state of the 3-state HMM corresponding to the triphone b-aa+g

- Training data corresponding to state j in phone p: Gather all x_t's that are tagged with ID *-p_j+*

How do we build these phone DTs?

1. What questions are used?

   Linguistically-inspired binary questions: “Does the left or right phone come from a broad class of phones such as vowels, stops, etc.?” “Is the left or right phone [k] or [m]?”

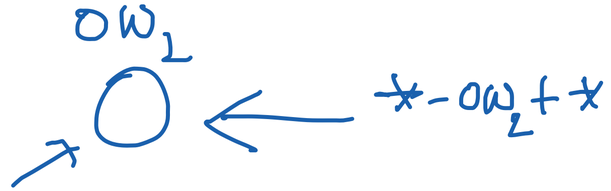
2. What is the training data for each phone state, p_j? (root node of DT)

   All speech frames that align with the j^{th} state of every triphone HMM that has p as the middle phone

3. What criterion is used at each node to find the best question to split the data on?

   Find the question which partitions the states in the parent node so as to give the maximum increase in log likelihood

Likelihood of a cluster of states



- If a cluster of HMM states, $S = \{s_1, s_2, \ldots, s_M\}$ consists of M states and a total of K acoustic observation vectors are associated with S, $\{x_1, x_2 \ldots, x_K\}$, then the log likelihood associated with S is:

$$\mathcal{L}(S) = \sum_{i=1}^{K} \sum_{s \in S} \log \Pr(x_i; \mu_S, \Sigma_S) \gamma_s(x_i)$$

state occupancy probability

- For a question q that splits S into S_{yes} and S_{no}, compute the following quantity:

$$\Delta_q = \mathcal{L}(S^q_{\text{yes}}) + \mathcal{L}(S^q_{\text{no}}) - \mathcal{L}(S)$$

$$q^* = \arg\max_q \Delta_q$$

Likelihood of a cluster of states

- If a cluster of HMM states, $S = \{s_1, s_2, \ldots, s_M\}$ consists of M states and a total of K acoustic observation vectors are associated with S, $\{x_1, x_2 \ldots, x_K\}$, then the log likelihood associated with S is:

$$\mathcal{L}(S) = \sum_{i=1}^{K} \sum_{s \in S} \log \Pr(x_i; \mu_S, \Sigma_S) \gamma_s(x_i)$$

- For a question q that splits S into S_{yes} and S_{no}, compute the following quantity:

$$\Delta_q = \mathcal{L}(S^q_{\text{yes}}) + \mathcal{L}(S^q_{\text{no}}) - \mathcal{L}(S)$$

- Go through all questions, find Δ_q for each question q and choose the question for which Δ_q is the biggest

Likelihood of a cluster of states

- If a cluster of HMM states, $S = \{s_1, s_2, \ldots, s_M\}$ consists of M states and a total of K acoustic observation vectors are associated with S, $\{x_1, x_2 \ldots, x_K\}$, then the log likelihood associated with S is:

$$\mathcal{L}(S) = \sum_{i=1}^{K} \sum_{s \in S} \log \Pr(x_i; \mu_S, \Sigma_S) \gamma_s(x_i)$$

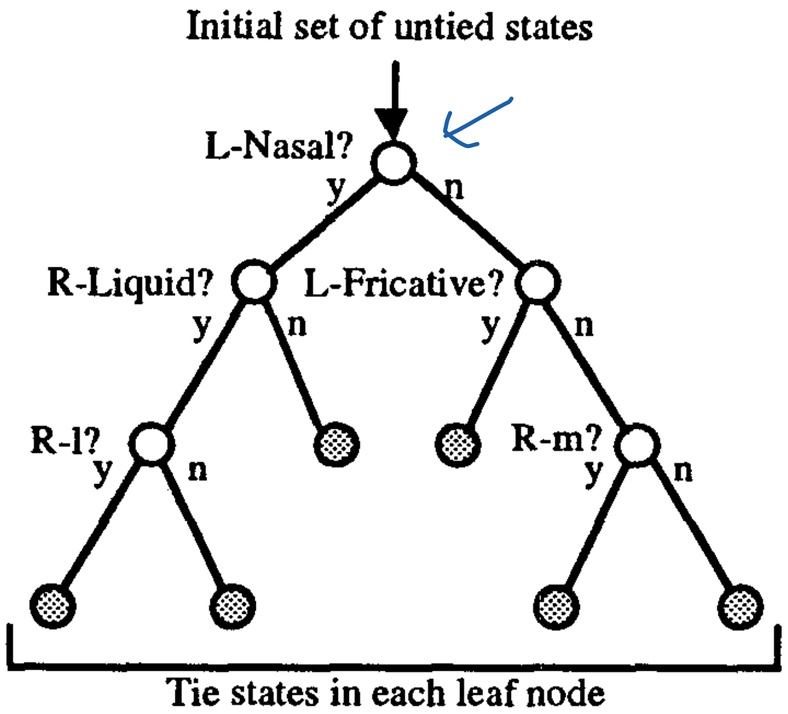
- For a question q that splits S into S_{yes} and S_{no}, compute the following quantity:

$$\Delta_q = \mathcal{L}(S^q_{\text{yes}}) + \mathcal{L}(S^q_{\text{no}}) - \mathcal{L}(S)$$

- Go through all questions, find Δ_q for each question q and choose the question for which Δ_q is the biggest

- Terminate when: Final Δ_q is below a threshold or data associated with a split falls below a threshold

Likelihood criterion



Given a phonetic question, let the initial set of untied states S be split into two partitions S_{yes} and S_{no}

Each partition is clustered to form a single Gaussian output distribution with mean $\mu_{S_{yes}}$ and covariance $\Sigma_{S_{yes}}$

Use the likelihood of the parent state and the subsequent split states to determine which question a node should be split on

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs

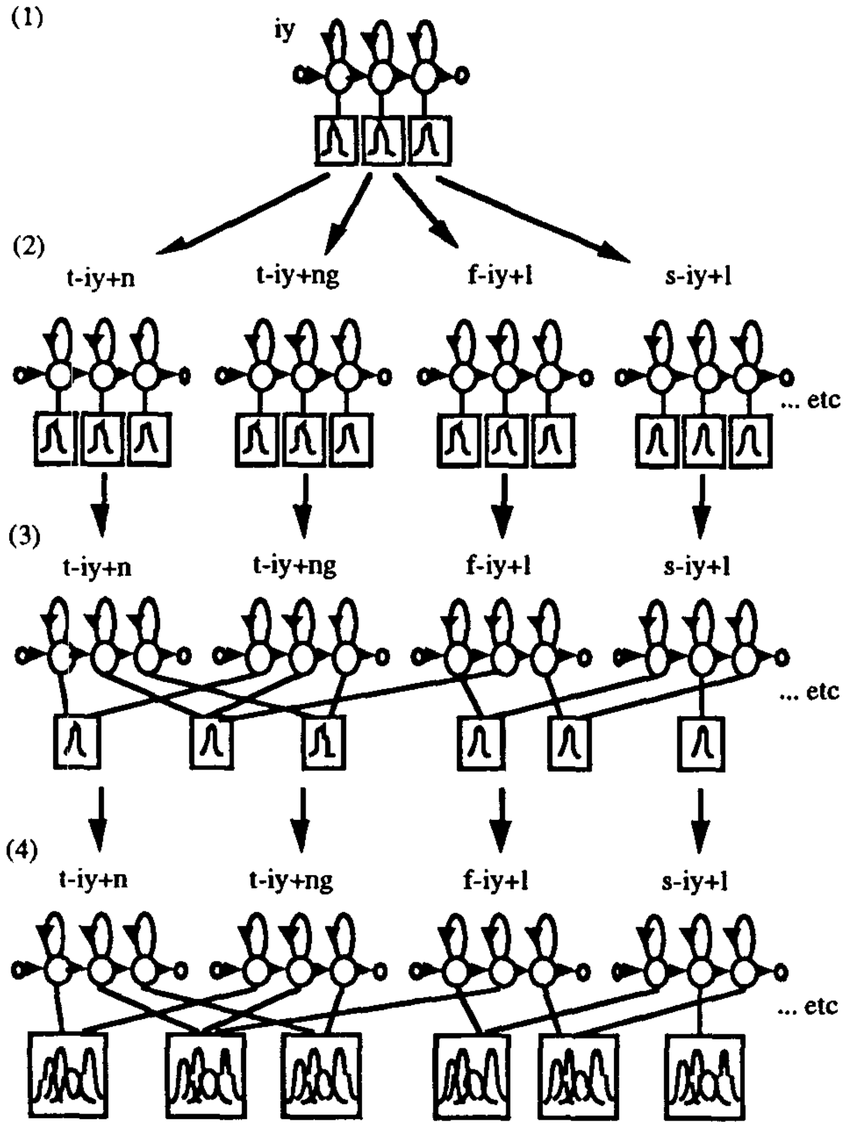
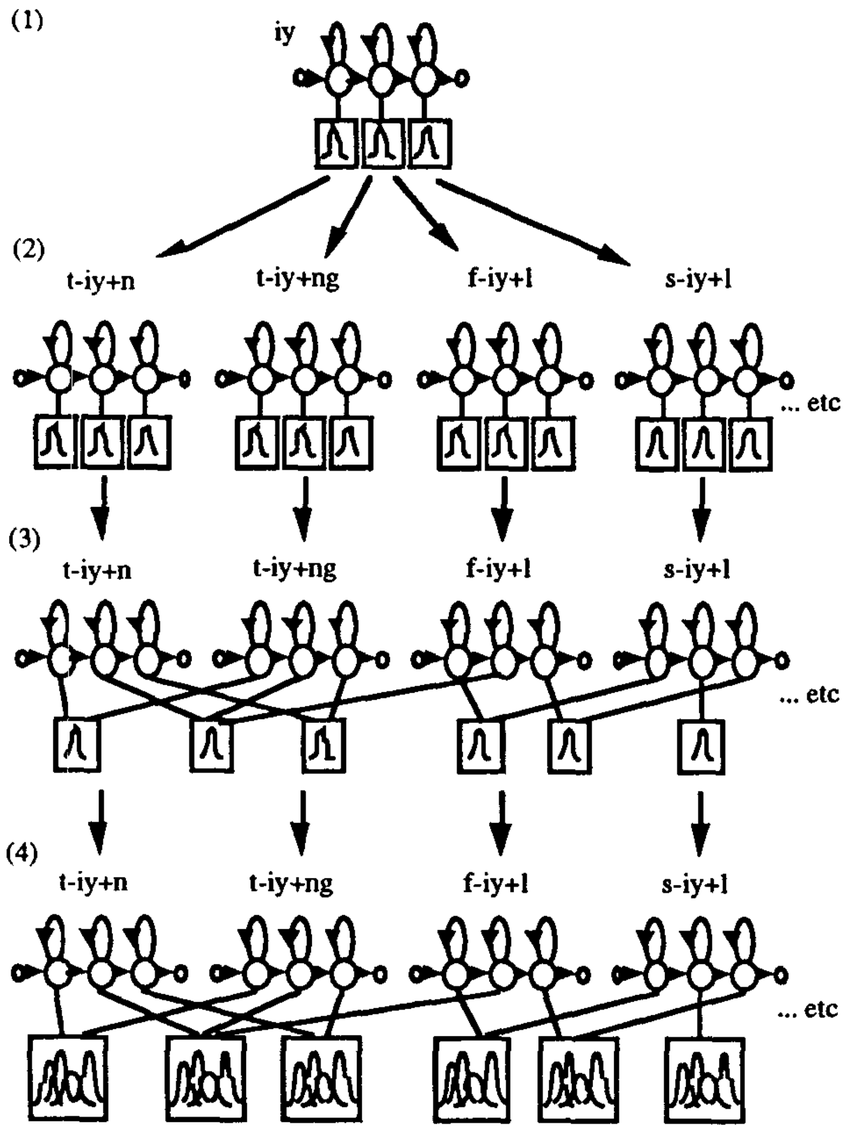


Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs

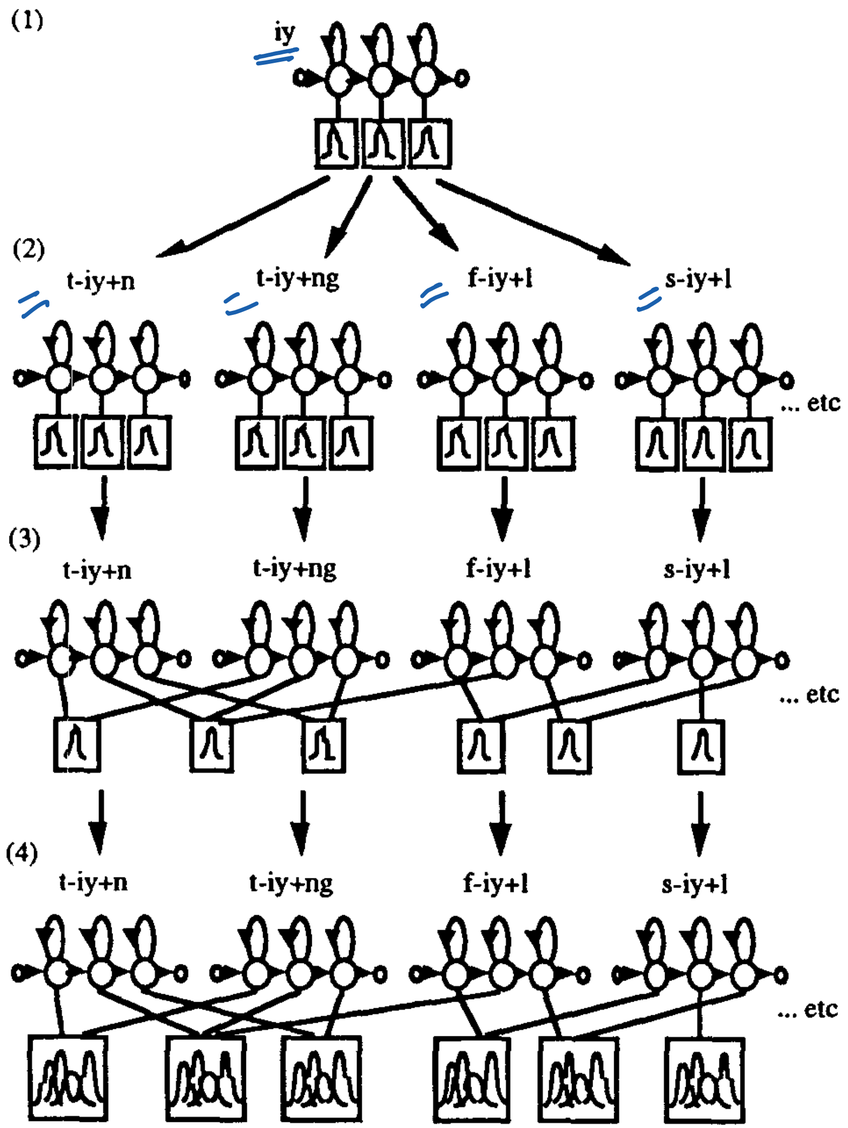


Four main steps in building a tied state HMM system:

1. Create and train 3-state monophone HMMs with single Gaussian observation probability densities

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs

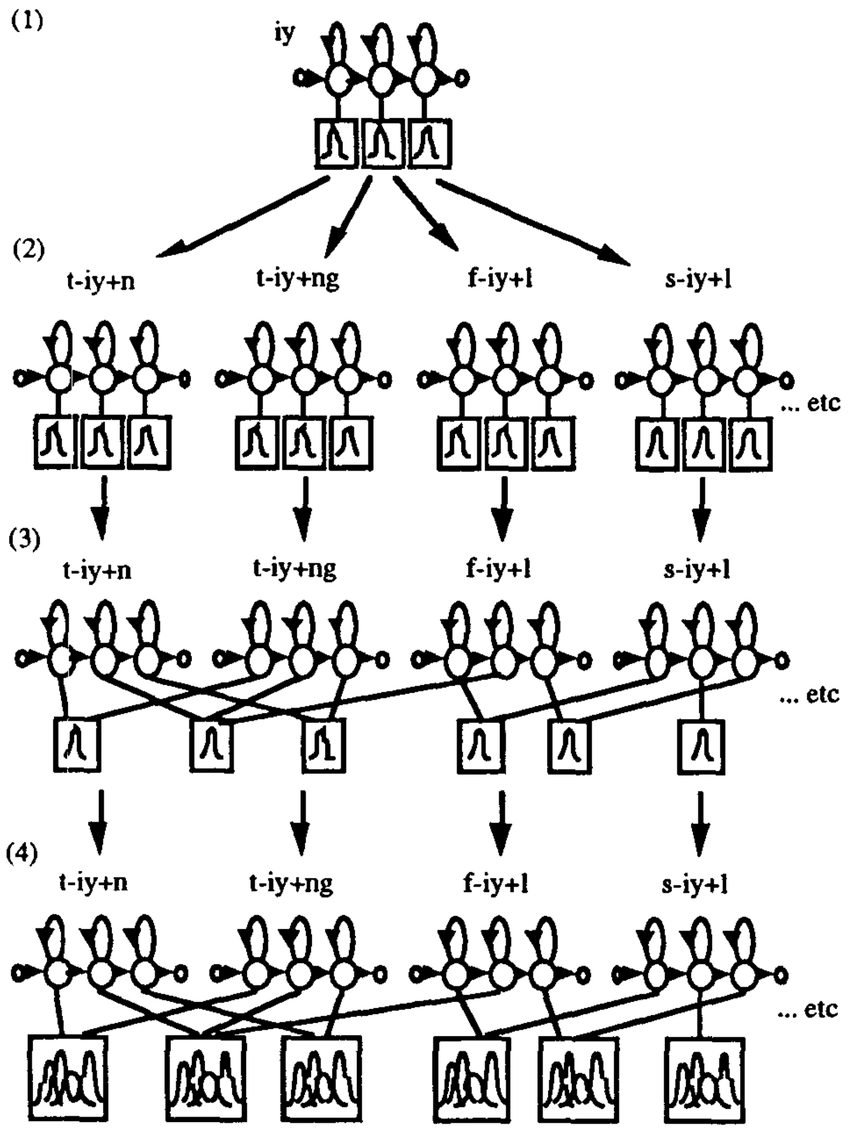


Four main steps in building a tied state HMM system:

1. Create and train 3-state monophone HMMs with single Gaussian observation probability densities
2. Clone these monophone distributions to initialise a set of untied triphone models. Train them using Baum-Welch estimation. Transition matrix remains common across all triphones of each phone.

Image from: Young et al., "Tree-based state tying for high accuracy acoustic modeling", ACL-HLT, 1994

Tied state HMMs

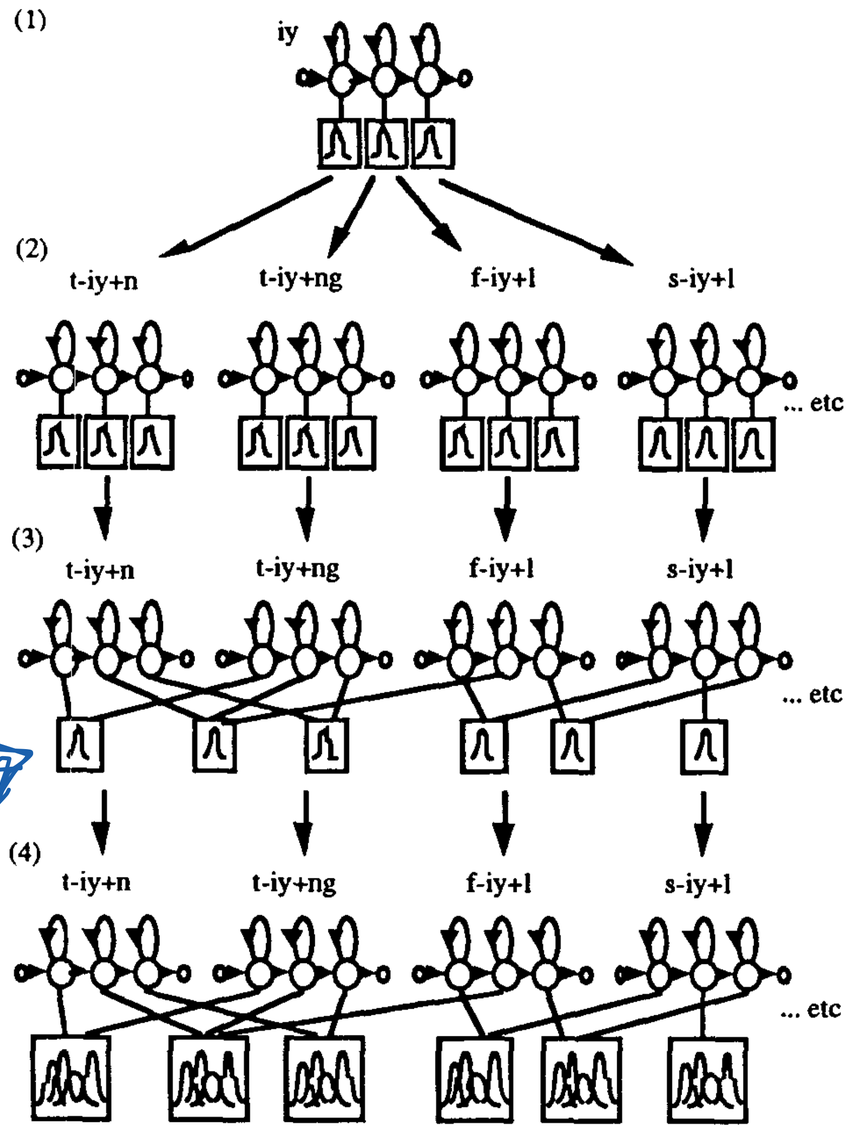


Four main steps in building a tied state HMM system:

1. Create and train 3-state monophone HMMs with single Gaussian observation probability densities
2. Clone these monophone distributions to initialise a set of untied triphone models. Train them using Baum-Welch estimation. Transition matrix remains common across all triphones of each phone.
3. For all triphones derived from the same monophone, cluster states whose parameters should be tied together.

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs



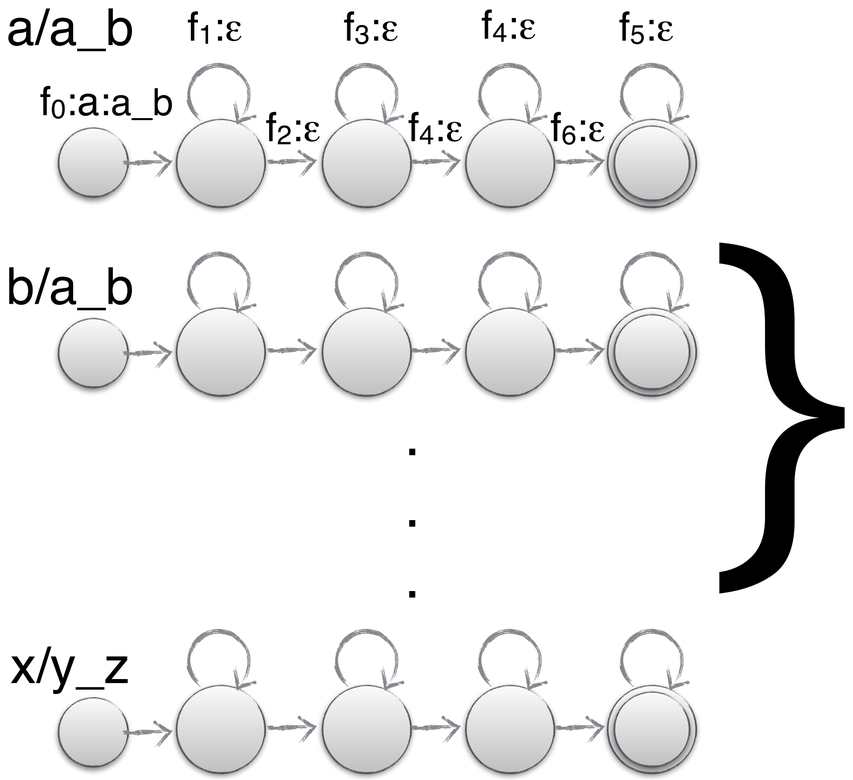
Four main steps in building a tied state HMM system:

1. Create and train 3-state monophone HMMs with single Gaussian observation probability densities
2. Clone these monophone distributions to initialise a set of untied triphone models. Train them using Baum-Welch estimation. Transition matrix remains common across all triphones of each phone.
3. For all triphones derived from the same monophone, cluster states whose parameters should be tied together.
4. Number of mixture components in each tied state is increased and models re-estimated using BW

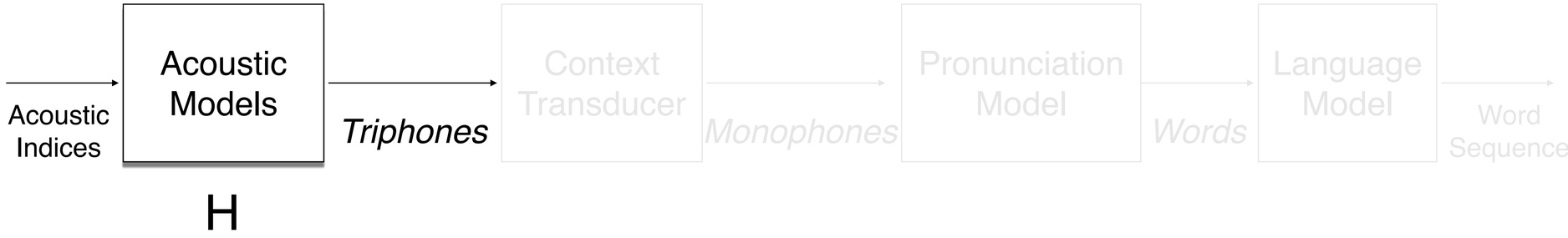
Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

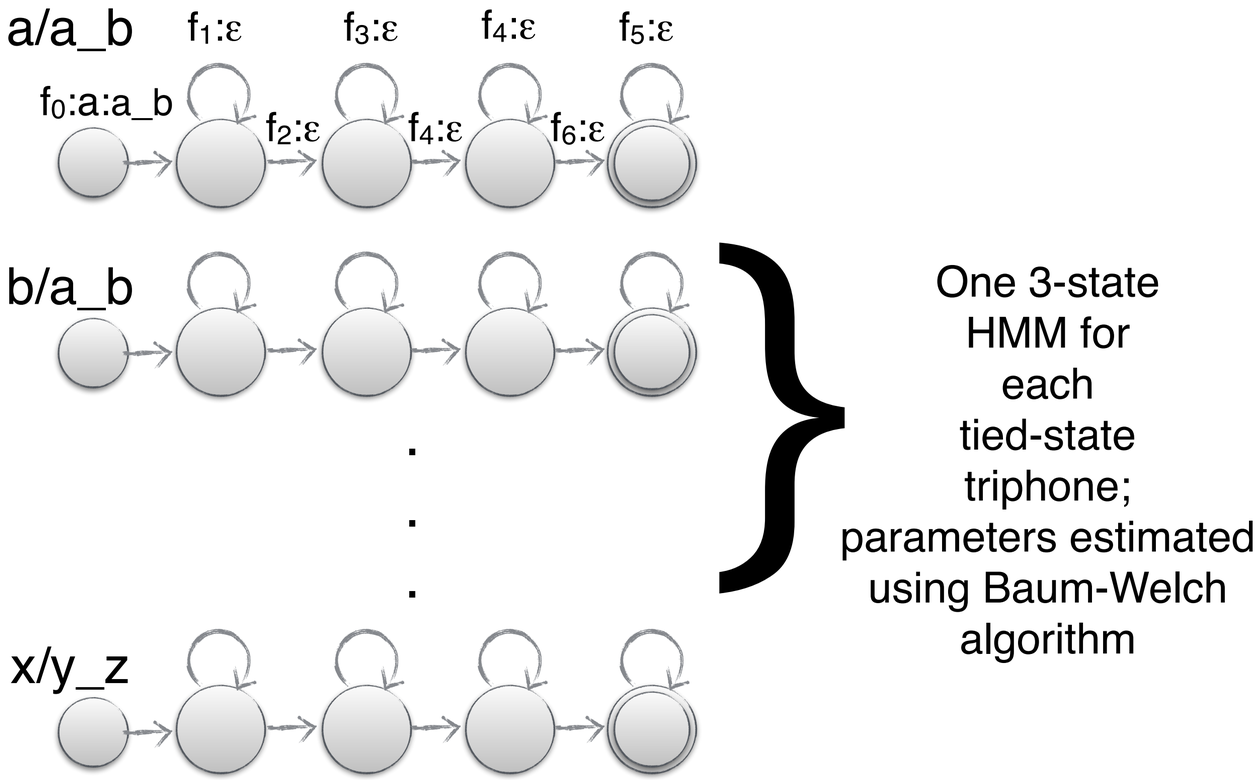
That's a wrap on HMM-based acoustic models

H



That's a wrap on HMM-based acoustic models

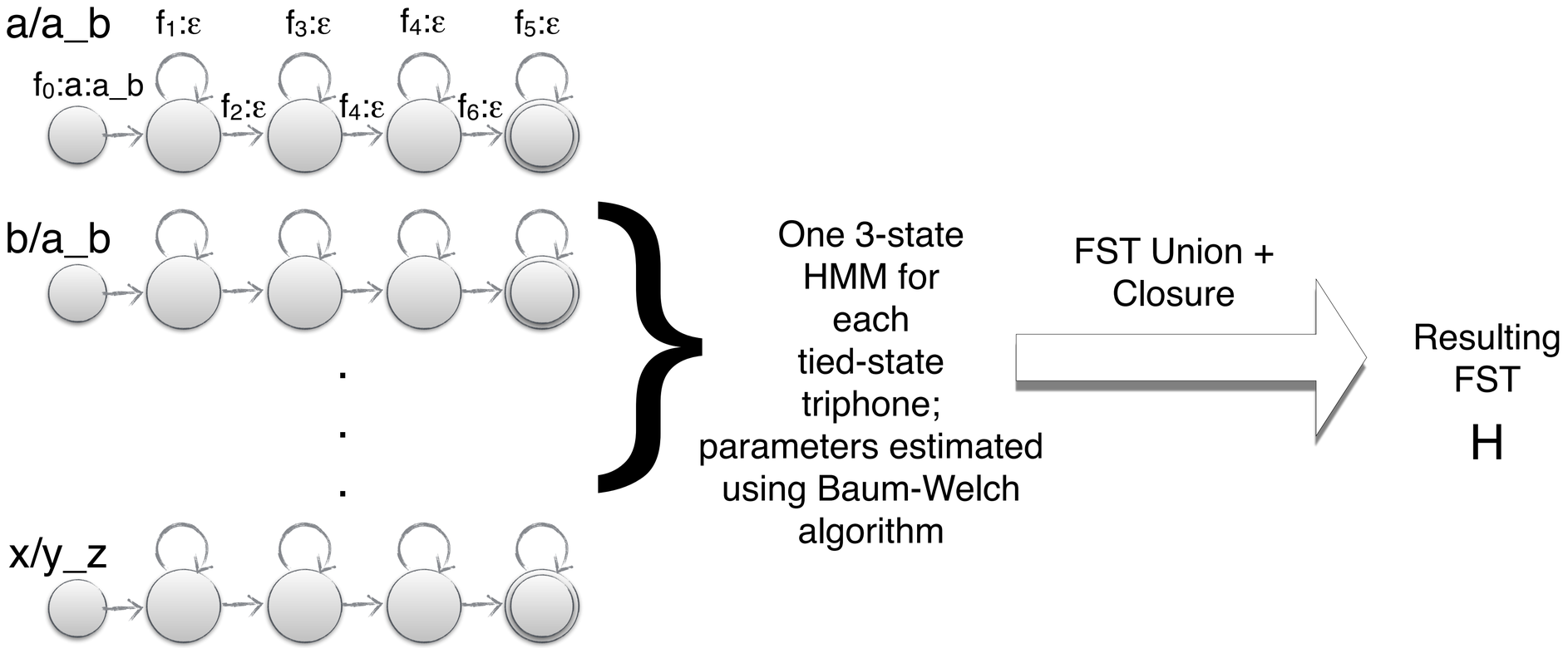




One 3-state HMM for each tied-state triphone; parameters estimated using Baum-Welch algorithm

That's a wrap on HMM-based acoustic models

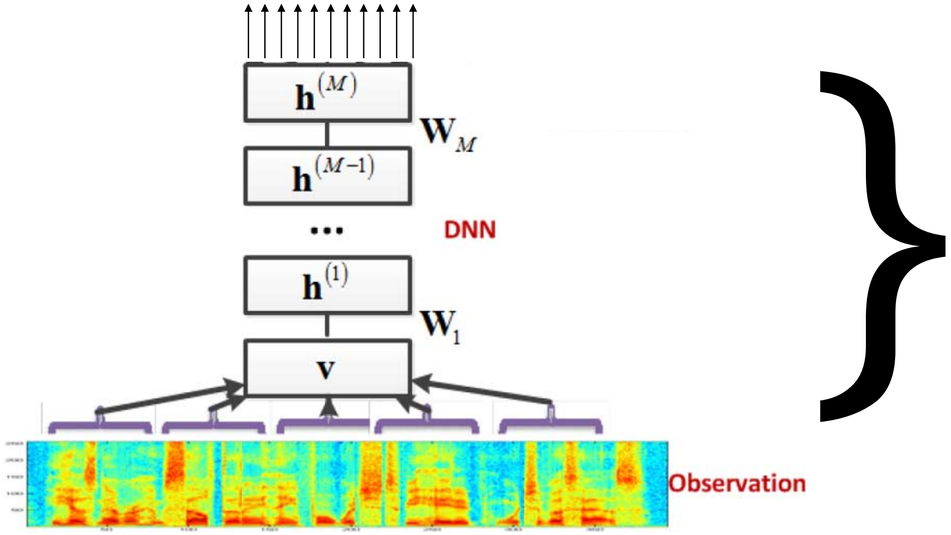
H



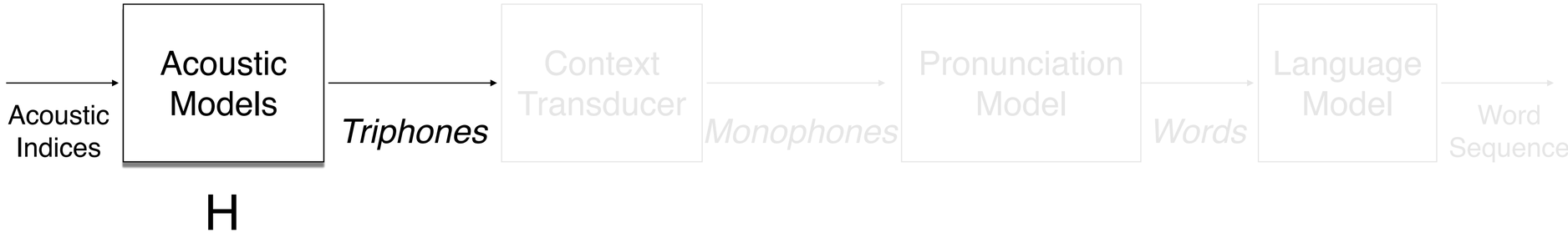
DNN-based acoustic models?

H

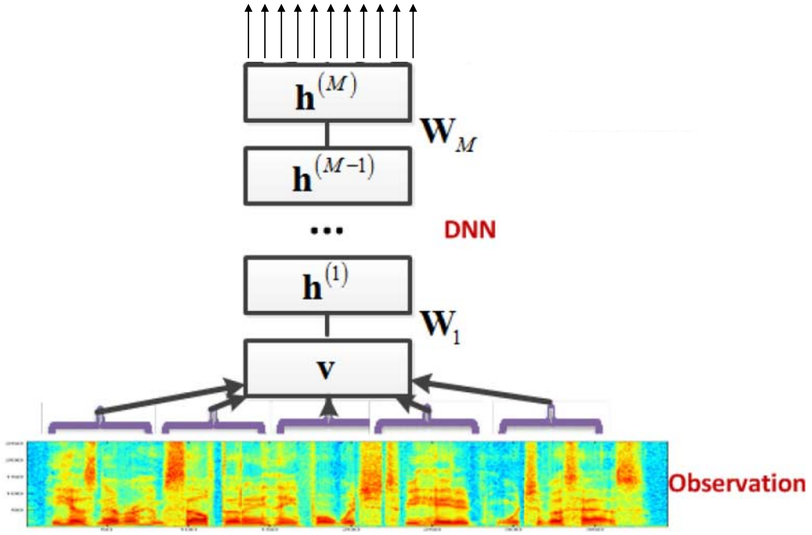
Phone posteriors



DNN-based acoustic models?

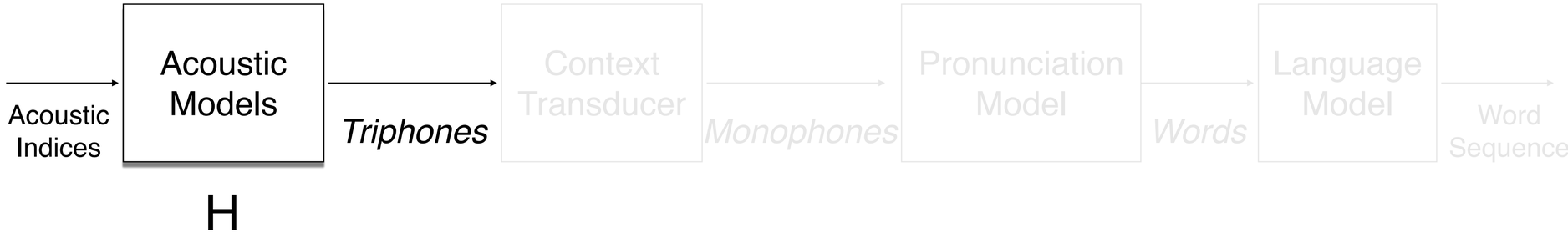


Phone posteriors

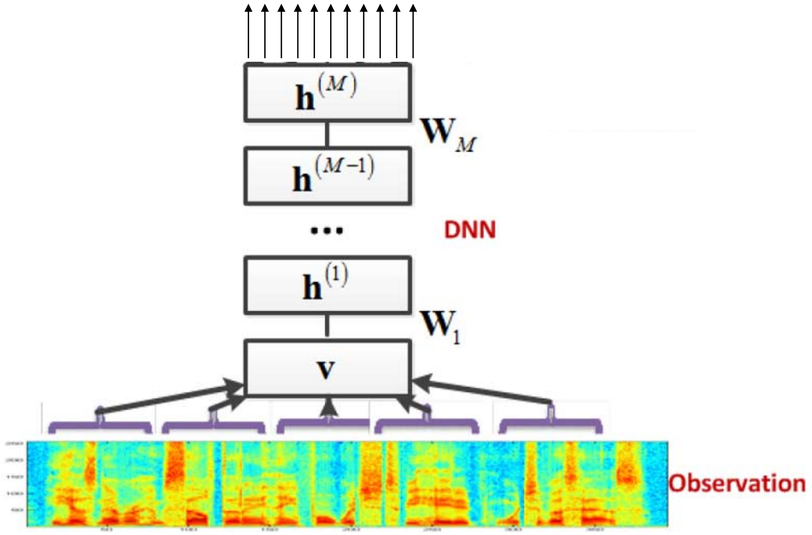


Can we use deep neural networks instead of HMMs to learn mappings between acoustics and phones?

DNN-based acoustic models?



Phone posteriors



Can we use deep neural networks instead of HMMs to learn mappings between acoustics and phones?

Resulting FST

H