Automatic Speech Recognition (CS753)

Lecture 8: Hidden Markov Models (IV) - Tied State Models

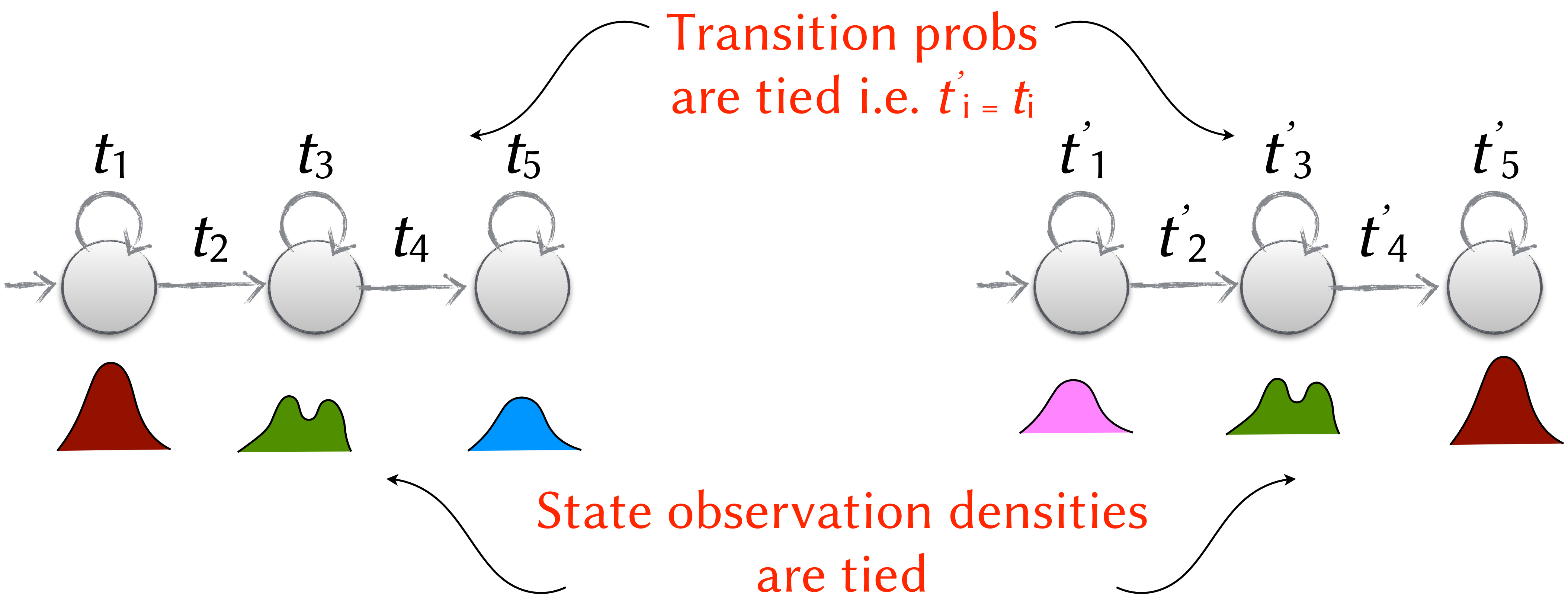
Instructor: Preethi Jyothi
Jan 30, 2017

Recap: 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

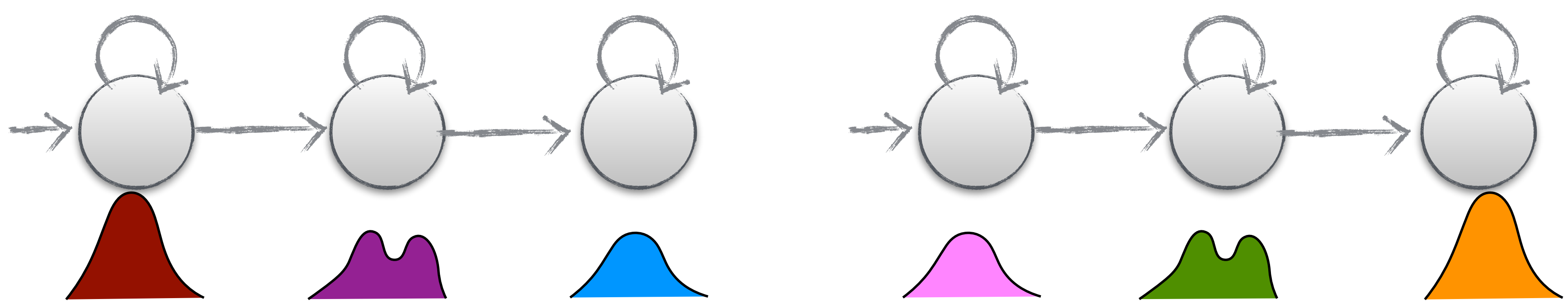
- Sharing of parameters (also referred to as "parameter tying") can be done at any level:
  - Parameters in HMMs corresponding to two triphones are said to be tied if they are identical



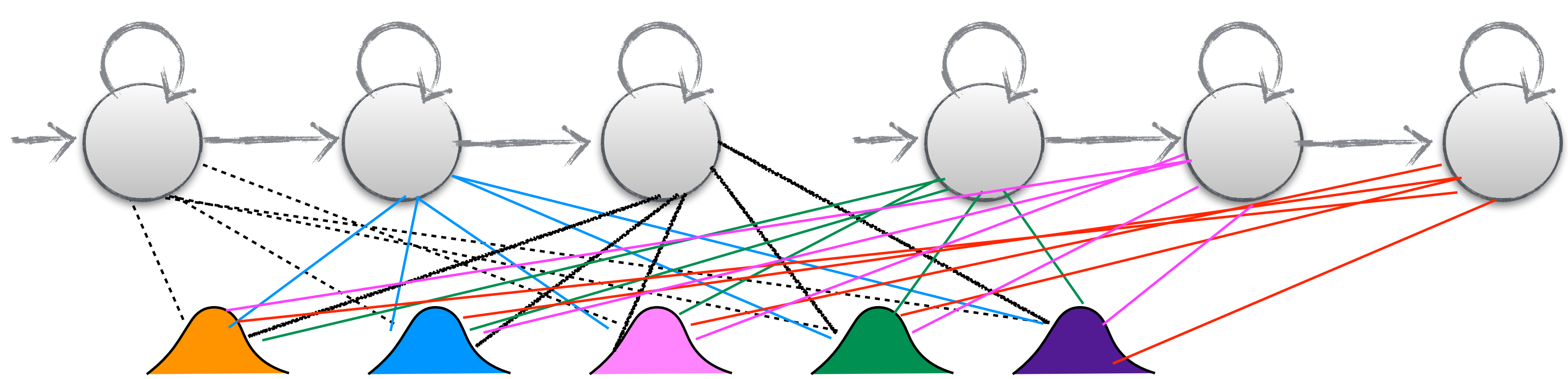
- More parameter tying: Tying variances of all Gaussians within a state, tying variances of all Gaussians in all states, tying individual Gaussians, etc.

1. Tied Mixture Models

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



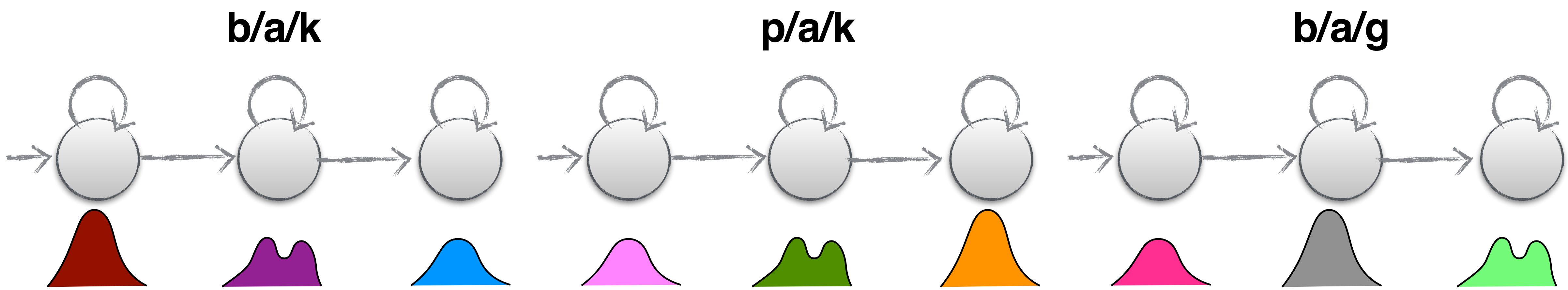
Triphone HMMs (No sharing)



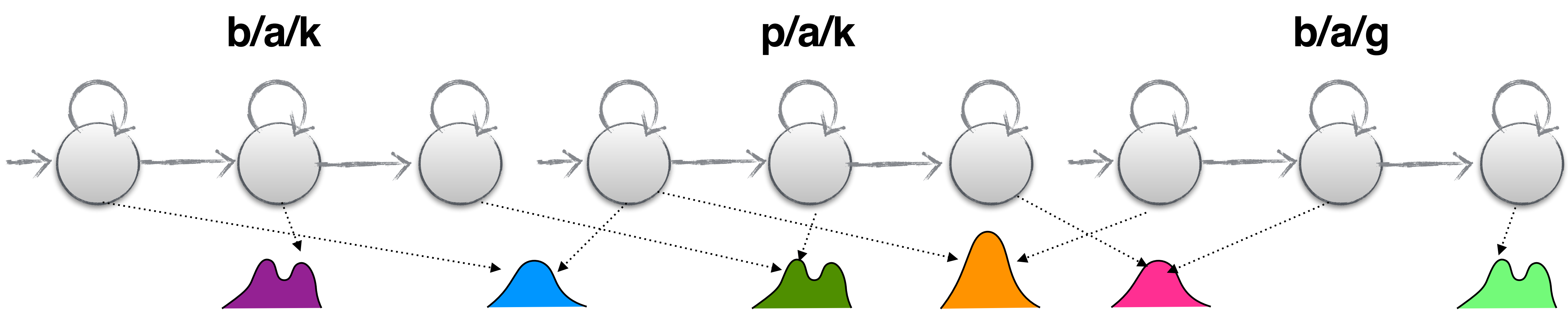
Triphone HMMs (Tied Mixture Models)

2. State Tying

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



Triphone HMMs (No sharing)



Triphone HMMs (State Tying)

Tied-state HMM system

Goal: Ensure there is sufficient training data to reliably estimate state observation densities while retaining important triphone distinctions

Three-steps:

1. Train HMM models (using Baum-Welch algorithm) without tying the parameters
2. Identify clusters of parameters which when tied together improve the model (i.e., increases the likelihood)
3. Tie together parameters in each identified cluster, and train the new HMM models (with fewer parameters)

Tied-state HMM system

Goal: Ensure there is sufficient training data to reliably estimate state observation densities while retaining important triphone distinctions

Three-steps:

1. Train HMM models (using Baum-Welch algorithm) without tying the parameters
2. Identify clusters of parameters which when tied together improve the model (i.e., increases the likelihood)
3. Tie t… HMM…

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

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

Tied-state HMM system

Goal: Ensure there is sufficient training data to reliably estimate state observation densities while retaining important triphone distinctions

Three-steps:

1. Train HMM models (using Baum-Welch algorithm) without tying the parameters
2. Identify clusters of parameters which when tied together improve the model (i.e., increases the likelihood)
3. Tie together parameters in each cluster, and train the new HMM models (with fewer parameters)

Number of mixture components within each tied state can be increased

Tied-state HMM system

<u>Goal</u>: Ensure there is sufficient training data to reliably estimate state observation densities while retaining important triphone distinctions

Three-steps:

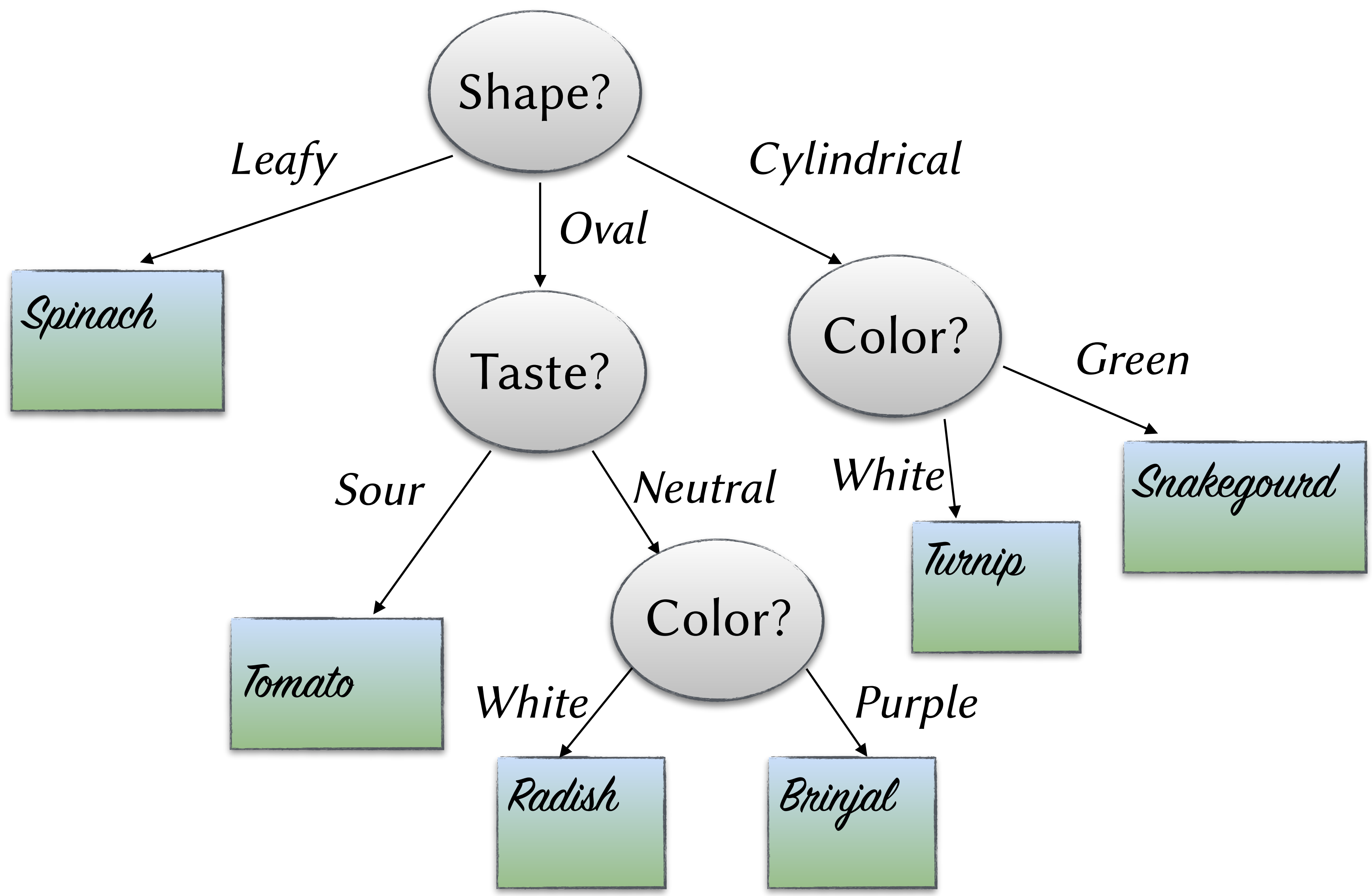
1. Train HMM models (using Baum-Welch algorithm) without tying the parameters
2. Identify clusters of parameters which when tied together improve the model (i.e., increases the likelihood)
3. Tie together parameters in each cluster, and train the new HMM models (with fewer parameters)

Try to optimize clustering, e.g., by learning a decision tree

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

- Given the data at a node, either declare the node to be a leaf or find another property to split the node into branches.

- Important questions to be addressed for DTs:
    1. How many splits at a node? Chosen by the user.
    2. Which property 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 HMM system

Goal: Ensure there is sufficient training data to reliably estimate state observation densities while retaining important context dependent distinctions

Three-steps:

1. Train HMM models (using Baum-Welch algorithm) without tying the parameters
2. Identify clusters of parameters which when tied together improve the model (i.e., increases the likelihood)
3. Tie together parameters in each cluster, and train the new HMM models (with fewer parameters)

Which parameters should be tied together? Use decision trees.

Top-down clustering
Phonetic Decision Trees

Build a decision tree for every state in every phone

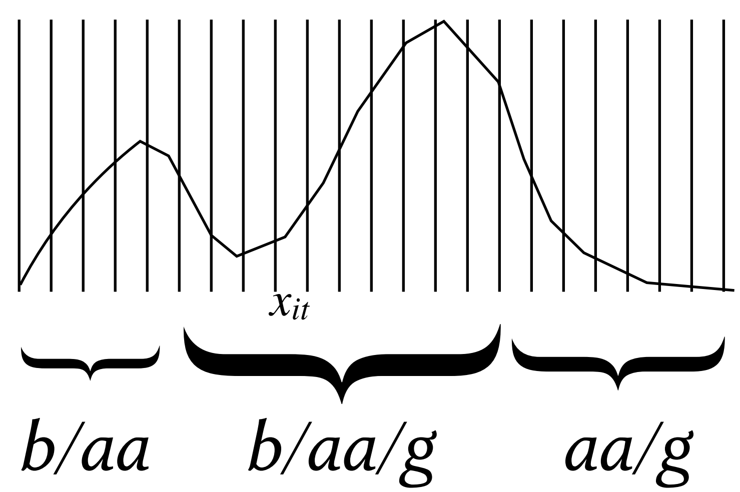
- For each phone p in { [ah], [ay], [ee], ... , [zh] }
  - For each state j in {0, 1, 2, ... }
    - Assemble training data corresponding to state j from all triphones with middle phone p (assumption about HMMs?)

Training data for DT nodes

- Align training data, $x_i = (x_{i1}, \ldots, x_{iT_i})$ $i=1\ldots N$ where $x_{it} \in \mathbb{R}^d$, against a set of triphone HMMs

- Use Viterbi algorithm to find the best HMM state sequence corresponding to each x_i

- Tag each x_{it} with ID of current phone along with left-context and right-context



x_{it} is tagged with ID aa_2[b/g] i.e. x_{it} is aligned with the second state of the 3-state HMM corresponding to the triphone b/aa/g

Top-down clustering

Phonetic Decision Trees

Build a decision tree for every state in every phone

- For each phone *p* in { [ah], [ay], [ee], … , [zh] }
  - For each state *j* in {0, 1, 2, … }
    - Assemble training data corresponding to state *j* from all triphones with middle phone *p*

Top-down clustering
Phonetic Decision Trees

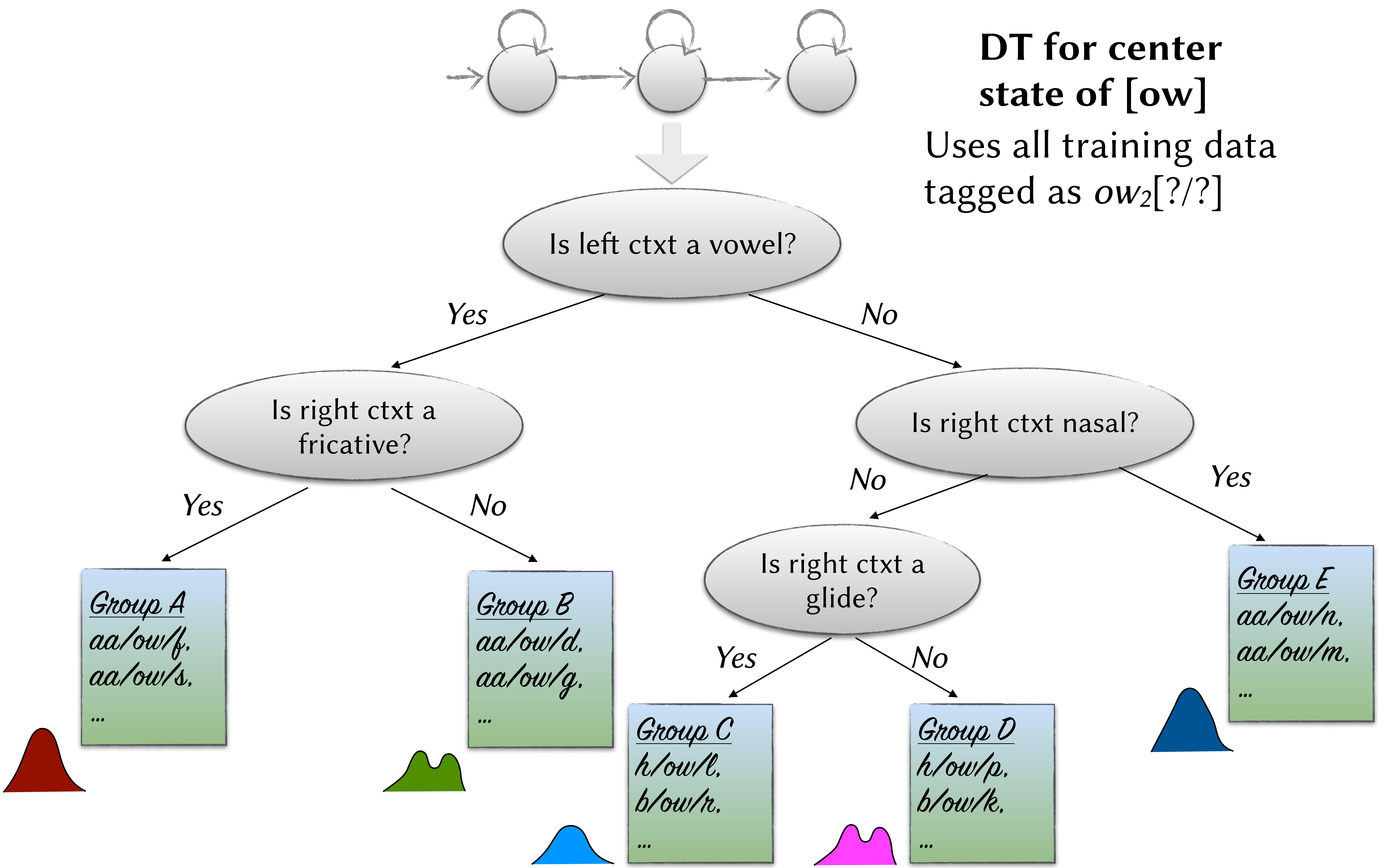
Build a decision tree for every state in every phone

- For each phone p in { [ah], [ay], [ee], … , [zh] }
    - For each state j in {0, 1, 2, … }
        - Assemble training data corresponding to state j from all triphones with middle phone p
        - Build a decision tree

Phonetic Decision Tree (DT)

DT for center state of [ow]

Uses all training data tagged as ow_2[?/?]



Top-down clustering
Phonetic Decision Trees

Build a decision tree for every state in every phone

- For each phone *p* in { [ah], [ay], [ee], ... , [zh] }
  - For each state *j* in {0, 1, 2, ... }
    - Assemble training data corresponding to state *j* from all triphones with middle phone *p*
    - Build a decision tree
    - Each leaf represents clusters of triphone models corresponding to state j

Top-down clustering
Phonetic Decision Trees

Build a decision tree for every state in every phone

- For each phone *p* in { [ah], [ay], [ee], … , [zh] }
  - For each state *j* in {0, 1, 2, … }
    - Assemble training data corresponding to state *j* from all triphones with middle phone *p*
    - Build a decision tree
    - Each leaf represents clusters of triphone models corresponding to state j
- If we have a total of N middle phones and each triphone HMM has *M* states, we will learn N * *M* decision trees

What phonetic questions are used?

- General place/manner of articulation related questions:
  - Stop: /k/, /g/, /p/, /b/, /t/, /d/, etc.
  - Fricative: /ch/, /jh/, /sh/, /s/, etc.
  - Vowel: /aa/, /ae/, /ow/, /uh/, etc.
  - Nasal: /m/, /n/, /ng/
- Vowel-based questions:
  - Front, back, central, long, diphthong, etc.
- Consonant-based questions:
  - Voiced or unvoiced, etc.
- **How do we choose the splitting question at a node?**

Choose splitting question based on likelihood measure

- Use likelihood of a cluster of states and of the subsequent splits to determine which question a node should be split on

- 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:

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

- If the output densities are Gaussian, then

$$\mathcal{L}(S) = -\frac{1}{2}(\log[(2\pi)^d |\Sigma_S|] + d) \sum_{i=1}^{K} \sum_{s \in S} \gamma_s(x_i)$$

Likelihood of a cluster of states

- Given a phonetic question, let S be split into two partitions S_{yes} and S_{no}

- Each partition is clustered to form a single Gaussian output distribution with mean μ_{Syes} and covariance Σ_{Syes}

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

State Splitting

- Likelihood of data after splitting on a yes/no question is given by:

$$\mathcal{L}(S_{\text{yes}}) + \mathcal{L}(S_{\text{no}})$$

- For a splitting question, compute the following quantity:

$$\Delta = \mathcal{L}(S_{\text{yes}}) + \mathcal{L}(S_{\text{no}}) - \mathcal{L}(\mathcal{S})$$

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

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

Overall process to construct a tied-state triphone HMM model

- Transition Matrix:
    - All triphones of a given phoneme use the same transition matrix common to all triphones of a phoneme
- State observation densities:
    - Use the triphone identity to traverse all the way to a leaf of the decision tree
    - Use the state observation probabilities associated with that leaf

Next class: Introduction to DNNs