Speech Recognition Architecture:

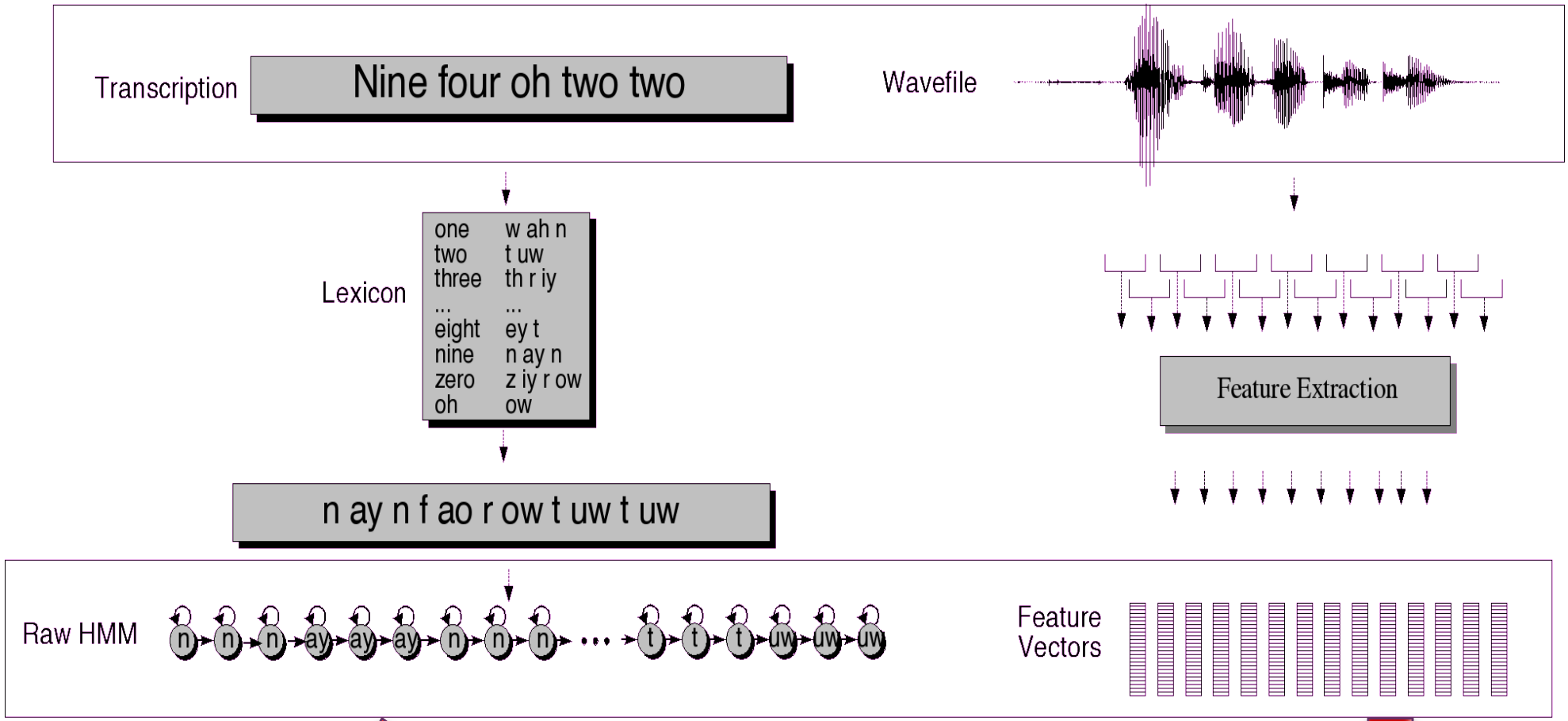
GMM Acoustic Models

OUCH: Outing the Unfortunate Characteristics of HMMs

CS 136a Speech Recognition
February 14, 2020
Professor Meteer

Thanks to Dan Jurafsky for these slides

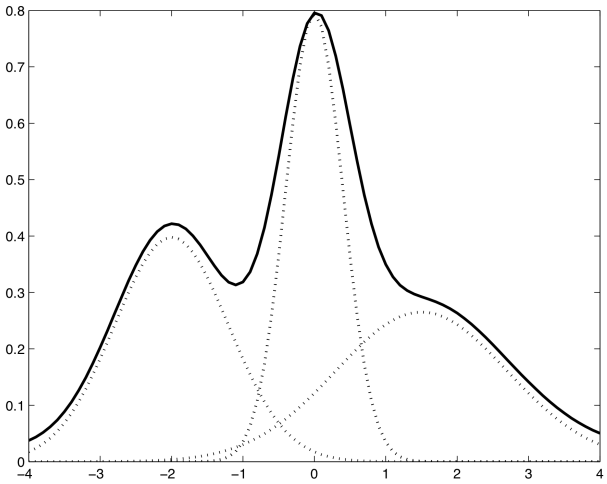
Back to Embedded Training



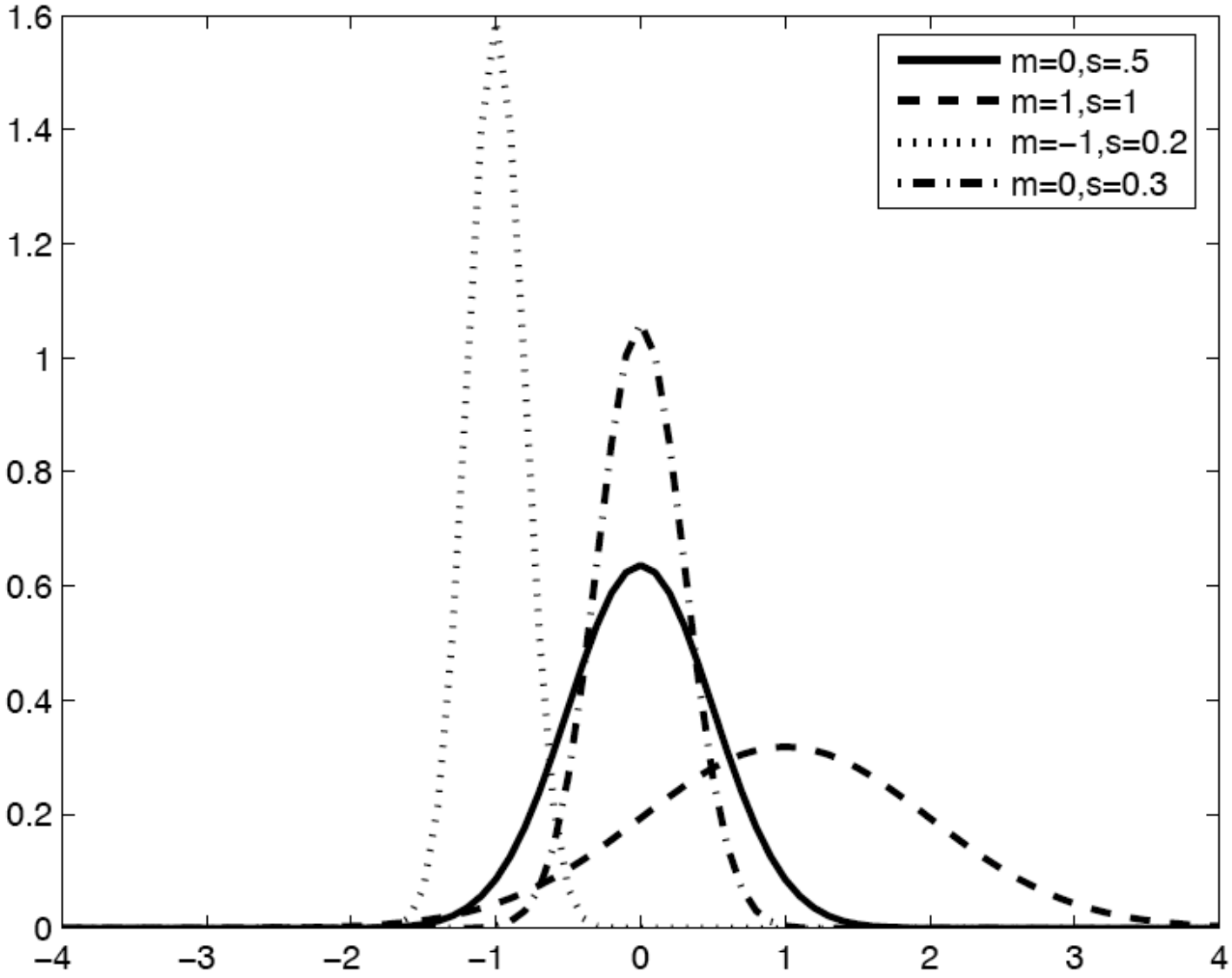
How do we model the observations?

Each element is the vector is a real value

- Multivariant Gaussian Mixture Models
  - **Gaussian**: Determine the likelihood of a real value to be in a particular state
  - **Multivariant**: We have a vector of values, not just one
  - **Mixture Models**: Values may not be best modeled by a single Gaussian
- Learning the “parameters” (means and variances) using the backward forward algorithm



Gaussians are parameters by mean and variance



Reminder: means and variances

- For a discrete random variable X
- Mean is the expected value of X
  - Weighted sum over the values of X

$$\mu = E(X) = \sum_{i=1}^{N} p(X_i) X_i$$

- Variance is the squared average deviation from mean

$$\sigma^2 = E(X_i - E(X))^2) = \sum_{i=1}^{N} p(X_i)(X_i - E(X))^2$$

+ BUT: We have a vector not a single value:

Multivariant Gaussians

- Gray-scale is real value from 0-100

Gray Scale

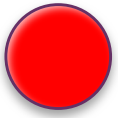
0 Black

50 Gray

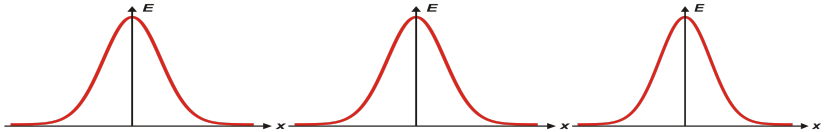
100 White

- Color is a combination of 3 values:

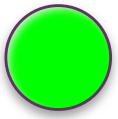
Color Vectors



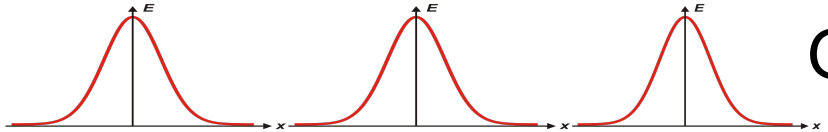
255	0	0



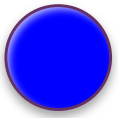
Red



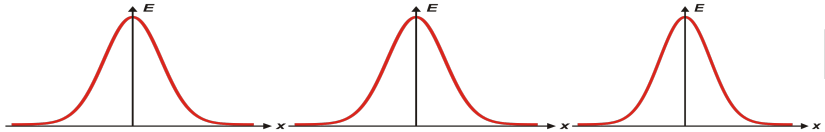
0	255	0



Green

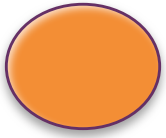
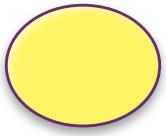
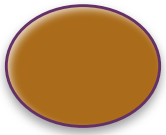
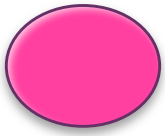
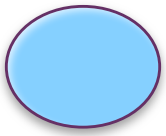


0	0	255



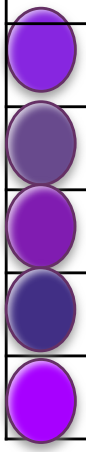
Blue

+ Example training data for color vectors

Observation				Label
243	142	53		Orange
255	244	102		Yellow
170	109	27		Brown
255	64	160		Pink
133	207	255		Blue

Learning “Purple” using Multivariant Gaussians

Collect all the observations labeled “purple”

	R	G	B
	135	38	224
	104	74	141
	128	28	177
	66	47	133
	167	0	255

Means: 120 37.4 186

Covariance matrix

	R	G	B
R	Var R	RG	RB
G	RG	Var G	GB
B	RB	GB	Var B

Are the elements of the vector independent?

Compare:

A lottery of 3 digits

If Independent: Each observation is modeled with two vectors: The mean and the diagonal of the covariance of the matrix

+ BUT Data is not always a single Gaussian

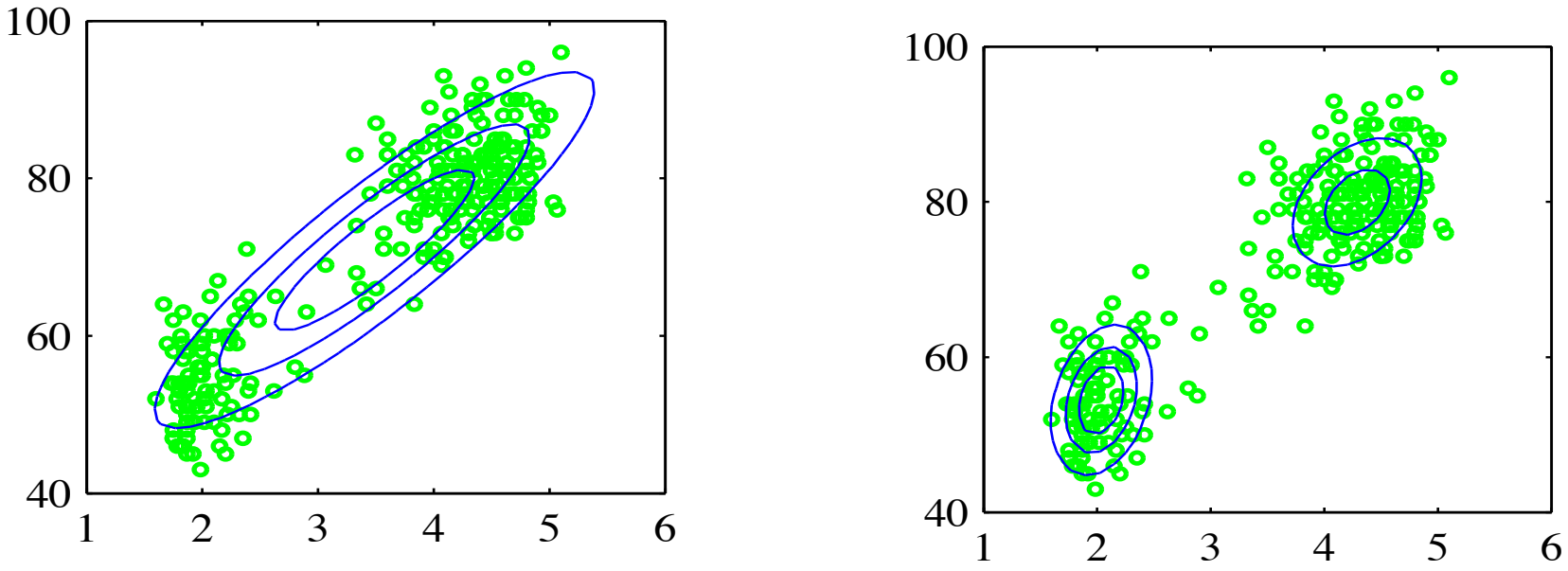
Gaussian Mixture Models

- Suppose you wanted to know the likely nationality of a student and all you knew was their height
  - Data: Height & Nationality
- Collect data
  - Each row is a student and their height and their nationality
  - Learn the mean and variance for each
  - “Decode”: For a new student, what’s the likelihood of being each nationality

Mixture models

- Suppose you find that the data does not fall into a nice Gaussian, but that if you model males and females separately, you have a better model
  - E.g. 5’8” is tall for a female but short for a male
- You can build a “mixture model” that better fits the data

Old Faithful Data



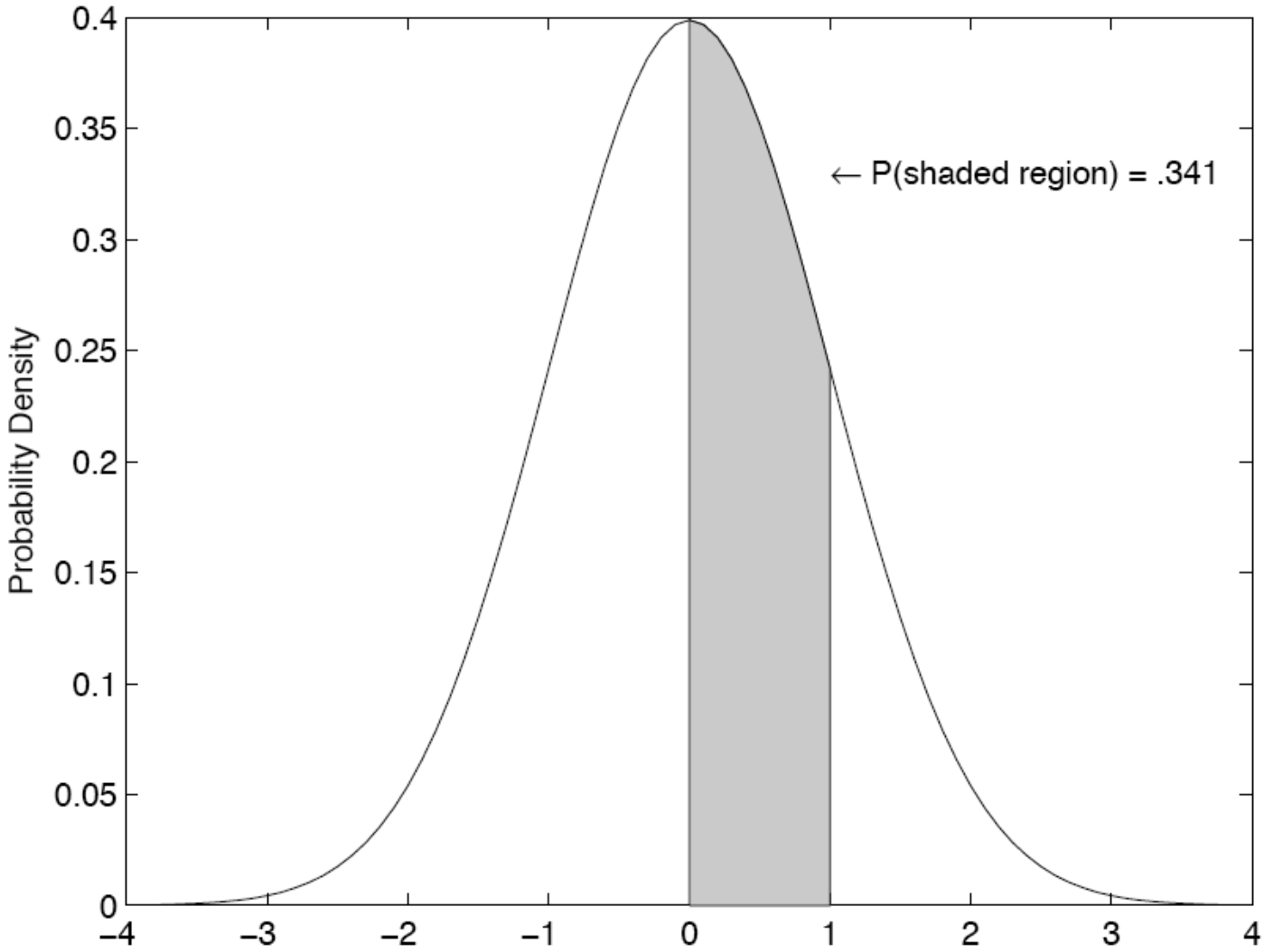
- Horizontal axis is duration of the eruption in minutes.
- Vertical axis is time until the next eruption in minutes.
- (a) A single Gaussian. (b) A mixture of two Gaussians.

Back to Acoustic Modeling

- Acoustic Model
  - Increasingly sophisticated models
  - Acoustic Likelihood for each state:
    - Gaussians
    - Multivariate Gaussians
    - Mixtures of Multivariate Gaussians
  - Where a state is progressively:
    - Context Independent Subphone (3ish per phone)
    - Context Dependent phone (triphones)
    - State-tying of Context Dependent phones

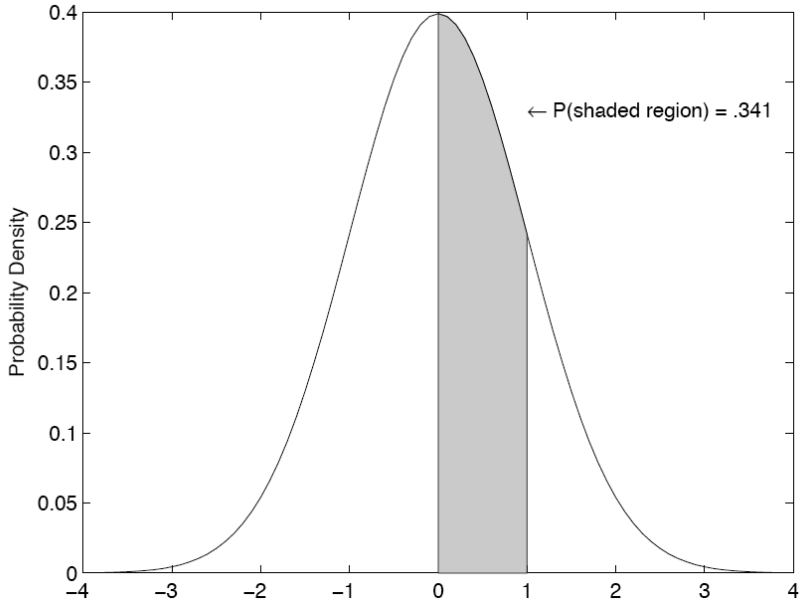
+ BUT What we really want is a probability

Gaussian as Probability Density Function



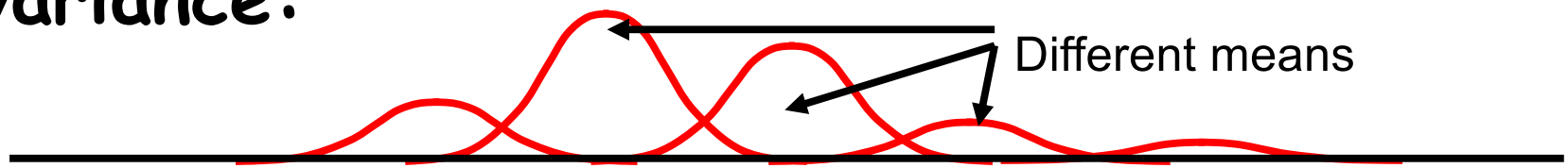
Gaussian PDFs

- A Gaussian is a probability density function; probability is the area under curve.
- To make it a probability, we constrain area under curve = 1
- BUT…
  - We will be using “point estimates”; value of Gaussian at point.
  - Technically these are not probabilities, since a pdf gives a probability over an interval, needs to be multiplied by dx
  - As we will see later, this is ok since the same value is omitted from all Gaussians, so argmax is still correct.

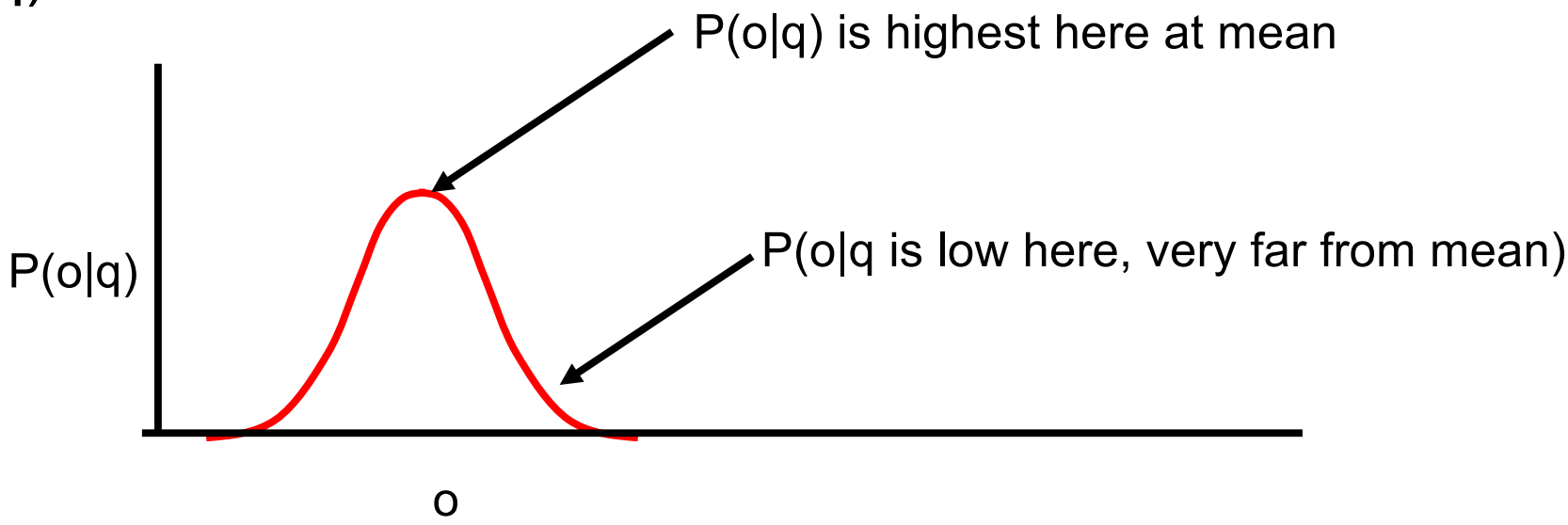


Gaussians for Acoustic Modeling

A Gaussian is parameterized by a mean and a variance:

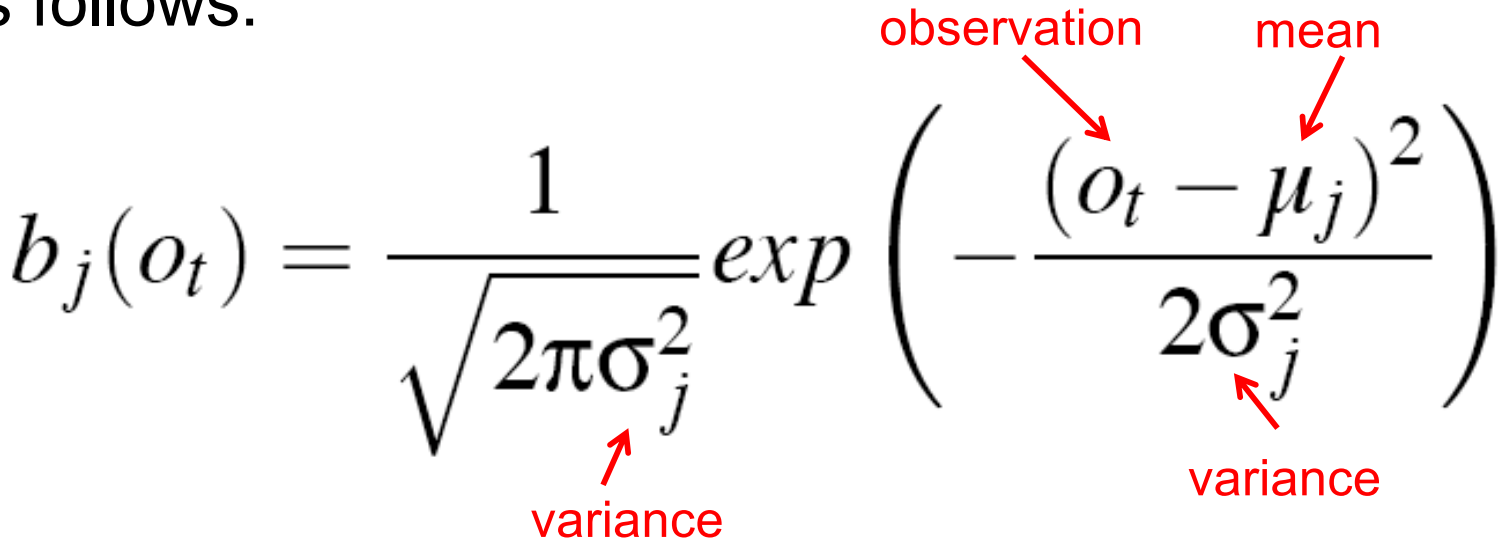


- P(o|q):



Using a (univariate) Gaussian as an acoustic likelihood estimator

- Let's suppose our observation was a single real-valued feature (instead of 39D vector)
- Then if we had learned a Gaussian over the distribution of values of this feature
- We could compute the likelihood of any given observation o_t as follows:



Multivariate Gaussians

- Instead of a single mean μ and variance σ:

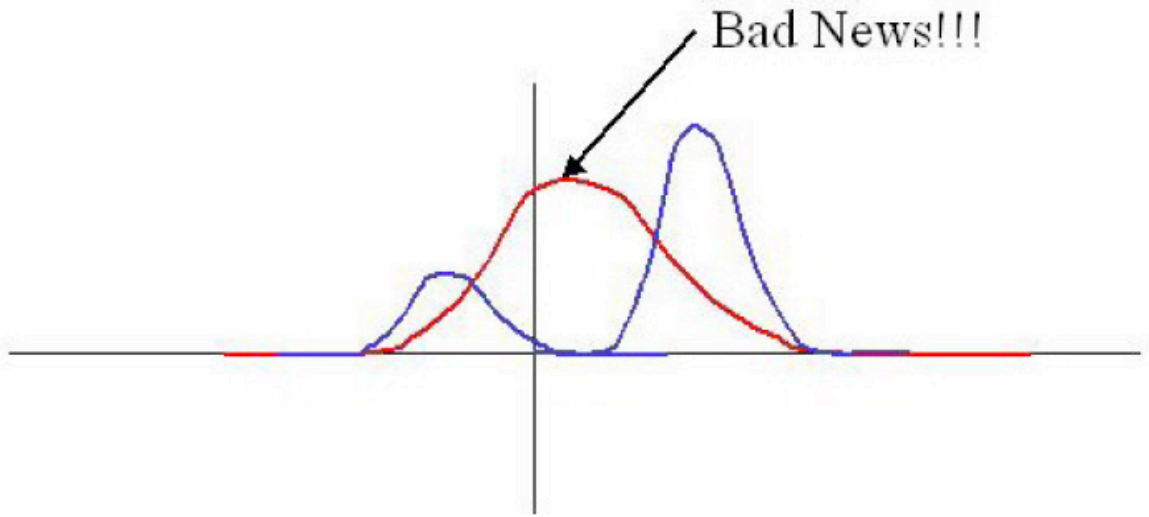
$$f(x \mid \mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

- Vector of observations *x* modeled by vector of means μ and covariance matrix Σ

$$f(x \mid \mu,\Sigma) = \frac{1}{(2\pi)^{D/2} \mid \Sigma \mid^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

But we're not there yet

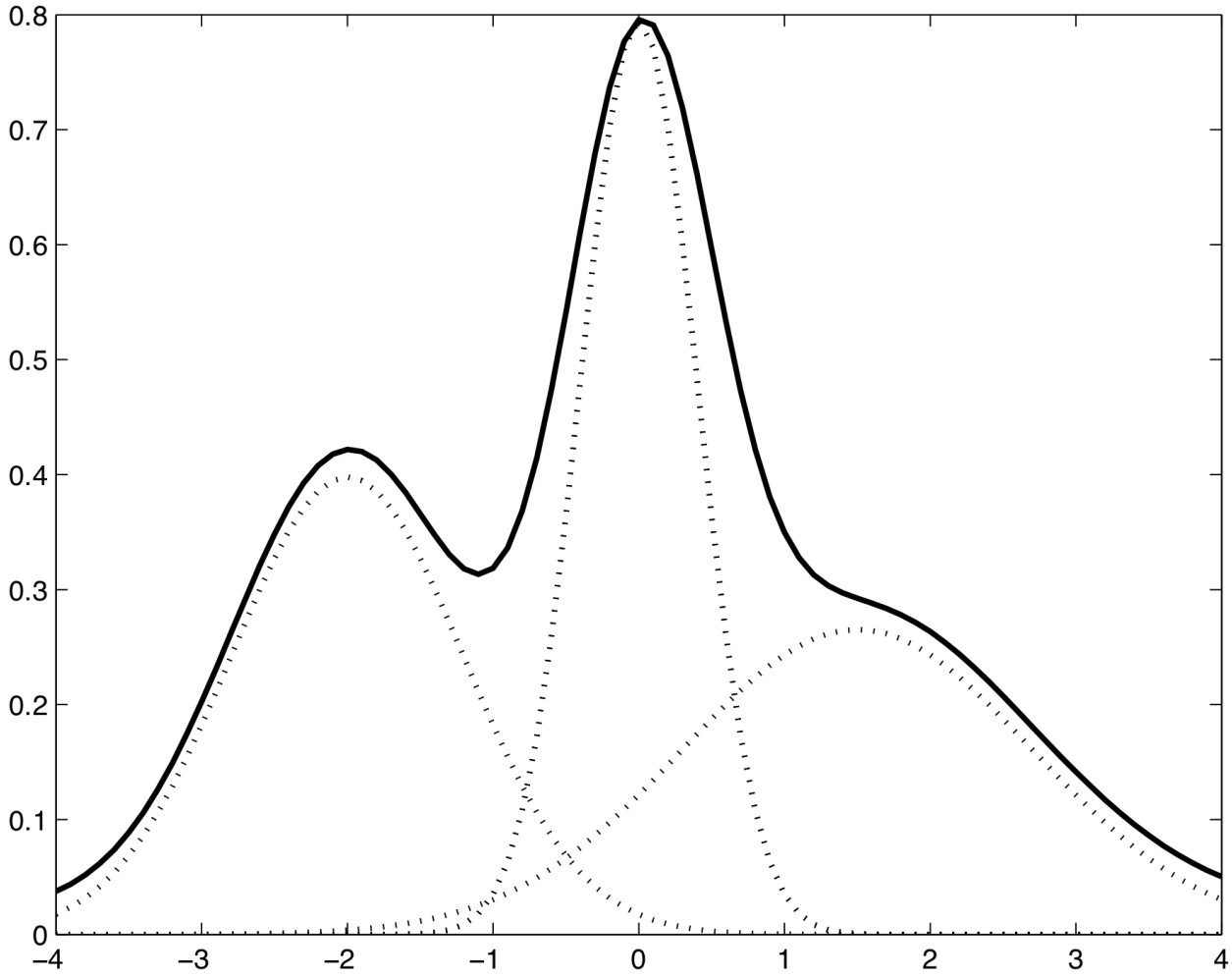
- Single Gaussian may do a bad job of modeling distribution in any dimension:



- Solution: Mixtures of Gaussians

Figure from Chen, Picheney et al slide

Mixture of Gaussians to model a function



Mixtures of Gaussians

- M mixtures of Gaussians:

$$f(x \mid \mu_{jk}, \Sigma_{jk}) = \sum_{k=1}^{M} c_{jk} \frac{1}{(2\pi)^{D/2} \mid \Sigma_{jk} \mid^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_{jk})^T \Sigma^{-1}(x-\mu_{jk})\right)$$

- For diagonal covariance:

$$b_j(o_t) = \sum_{k=1}^{M} c_{jk} \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{jkd}^2}} \exp\left(-\frac{1}{2}\left(\frac{o_{td} - \mu_{jkd}}{\sigma_{jkd}}\right)^2\right)$$

$$b_j(o_t) = \sum_{k=1}^{M} \frac{c_{jk}}{2\pi^{D/2} \prod_{d=1}^{D} \sigma_{jkd}^{\ 2}} \exp\left(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_{jkd} - \mu_{jkd})^2}{\sigma_{jkd}^{\ 2}}\right)$$

GMMs

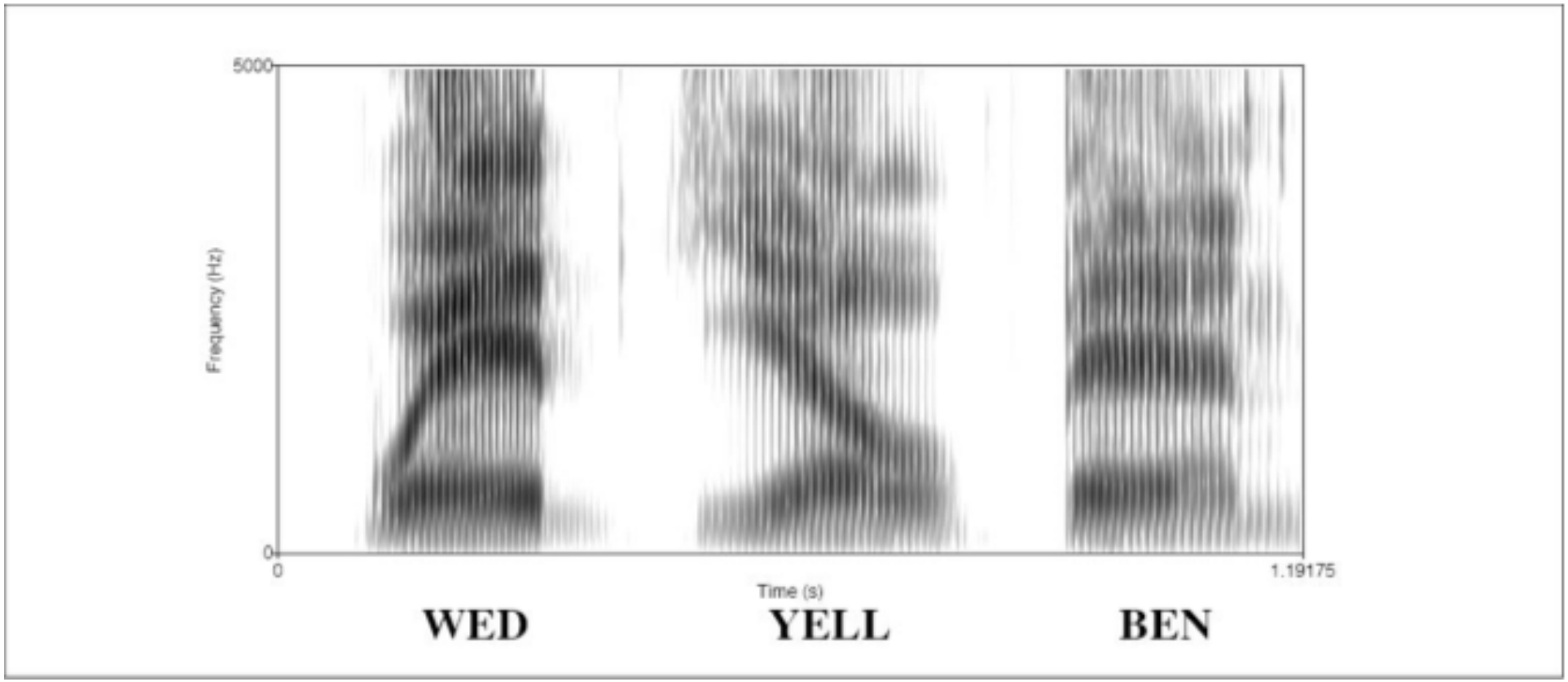
- Summary: each state has a likelihood function parameterized by:
  - M Mixture weights
  - M Mean Vectors of dimensionality D
  - Either
    - M Covariance Matrices of DxD
  - Or more likely
    - M Diagonal Covariance Matrices of DxD
      - which is equivalent to
      - M Variance Vectors of dimensionality D

Context Dependent Acoustic Models: Triphones

- Our phoneme models represent each phones with 3 states: beginning middle and end
- But rather then just modeling the phonemes, we model the phonemes in context
- A “Triphone” model represents a phone with a particular right and left context.

Thanks to Dan Jurafsky for these slides

+ Phoneme Variation



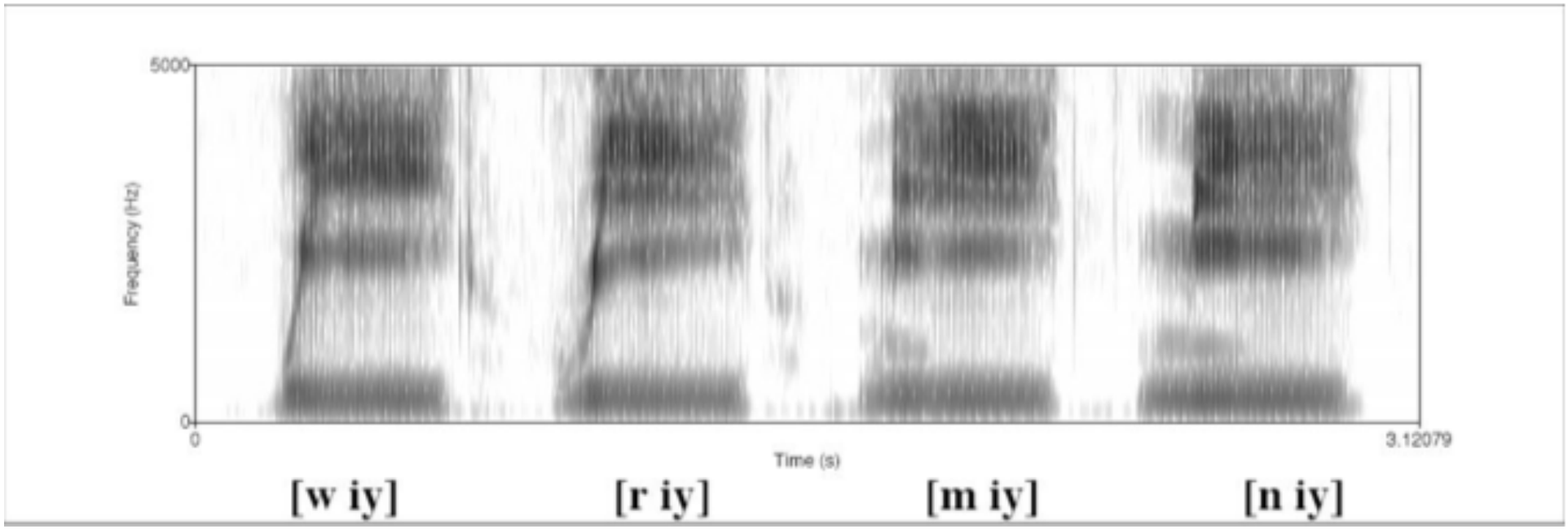
Thanks to Dan Jurafsky for these slides

Sparse data problem

- For a 50 phoneme set we would need 125,000 triphones
  - In practice, not all combinations occur
    - 55K triphones needed for 20K word WSJ corpus
    - Only 18.5K occurred in the training data
  - Attempting to train all of these triphones would result in many of then not having enough samples to adequately train.
- Reducing triphone parameters
  - Clustering contexts similar contexts
  - Tying **subphones** whose clusters fall into the same contexts
  - States that are “shared” use the same Gaussians
  - This significantly cuts down on the number of parameters to be trained

Thanks to Dan Jurafsky for these slides

+ Phonemes with similar contexts



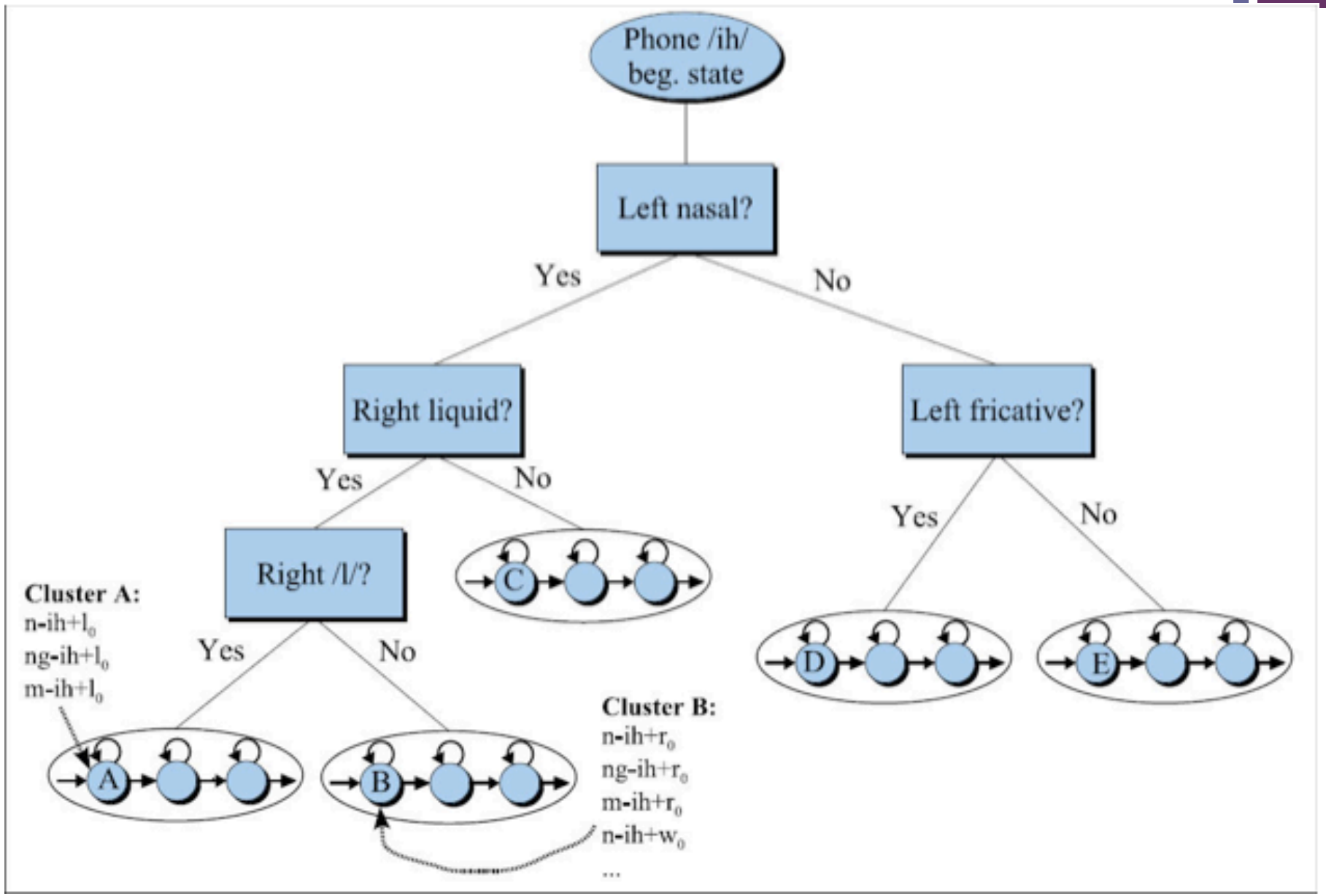
Thanks to Dan Jurafsky for these slides

How to determine which contexts to cluster?

- Decision tree based on phonetic features
- Root is the phoneme with all contexts
- Each level of the tree splits the cluster based on a set of questions about a particular phonetic features
  - Generally based on articulatory features

Feature	Phones
Stop	b d g k p t
Nasal	m n ng
Fricative	ch dh f jh s sh th v z zh
Liquid	l r w y
Vowel	aa ae ah ao aw ax axr ay eh er ey ih ix iy ow oy uh uw
Front Vowel	ae eh ih ix iy
Central Vowel	aa ah ao axr er
Back Vowel	ax ow uh uw
High Vowel	ih ix iy uh uw
Rounded	ao ow oy uh uw w
Reduced	ax axr ix
Unvoiced	ch f hh k p s sh t th
Coronal	ch d dh jh l n r s sh t th z zh

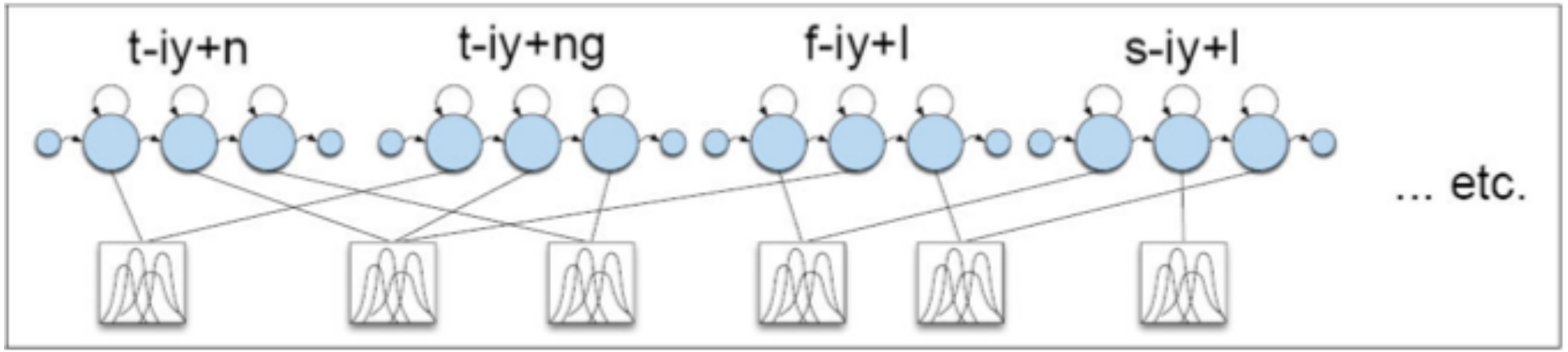
Decision Tree



Thanks to Dan Jurafsky for these slides

Tied states

a: t-iy-n
b: t-iy-ng
c: f-iy-l
d: s-iy-l



a1 a2 a3 b1 b2 b3 c1 c2 c3 d1 d2 d3 **12 states**

ab1 abc2 ab3 ab1 abc2 ab3 cd1 abc2 c3d cd1 d2 cd3 **6 states**

Thanks to Dan Jurafsky for these slides

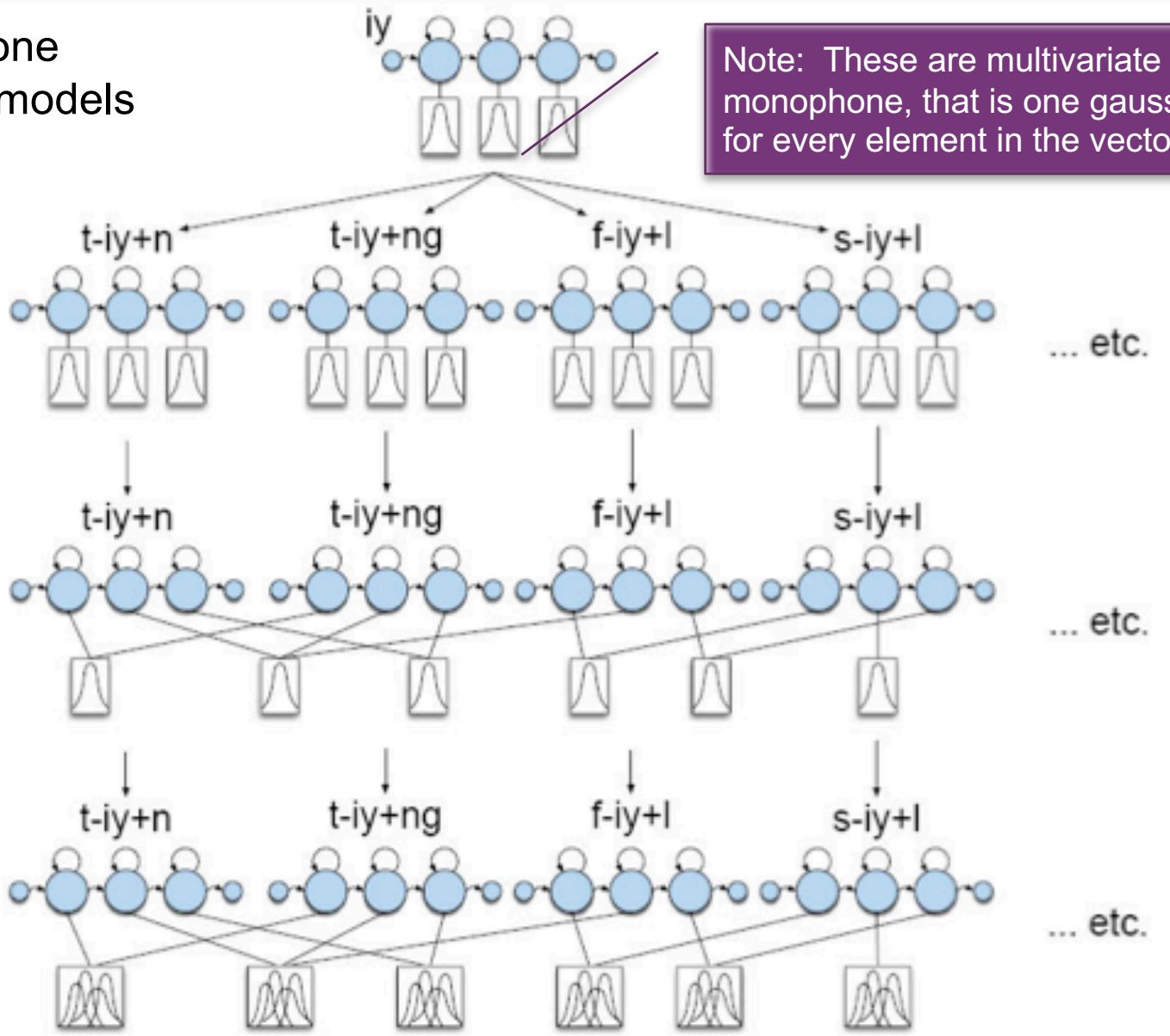
Steps to train Continuous Density State Tied models

1. Train monophone single Gaussian models

2. Clone monophones to triphones

3. Cluster and tie triphones

4. Expand to GMMs



Note: These are multivariate monophone, that is one gaussians for every element in the vector

Thanks to Dan Jurafsky for these slides

What's wrong with Acoustic Models?

- OUCH: Outing the Unfortunate Characteristics of HMMs

“Independence” Assumptions in AM

- Transition probabilities are independent from each other
  - Hidden under Markov blanket.
- Emission probabilities are independent from each other
  - Each observation is conditioned on only one state.
- A and B are conditionally independent
  - Stationarity, at transition from $q_{i,t}$ to $q_{j,t+1}$, its probability $\alpha_{i->j}$ is independent no matter what observation, o_t is conditioned on $q_{i,t}$.
- Observations are in multivariate normal distribution with diagonal covariance
  - Remember that if Cov(x,y) == 0: $x \perp y$, thus, by ignoring non-diagonals , we treat all features as independent from each other.

Independence “Assumptions” in AM

- We don’t know these conditional independences hold in real speech data, *we just assume*.
- What if we have a dataset that satisfies, *for 100% sure*, the independences?
  - **If HMM works differently (presumably better) with that data than real speech data, it proves that these independence assumptions on real speech are wrong**.
    (Classic form of proof by contradiction)
- How can we get this particular data?

  ➔ We use artificial data stochastically simulated.

+ Sources for Data Simulation

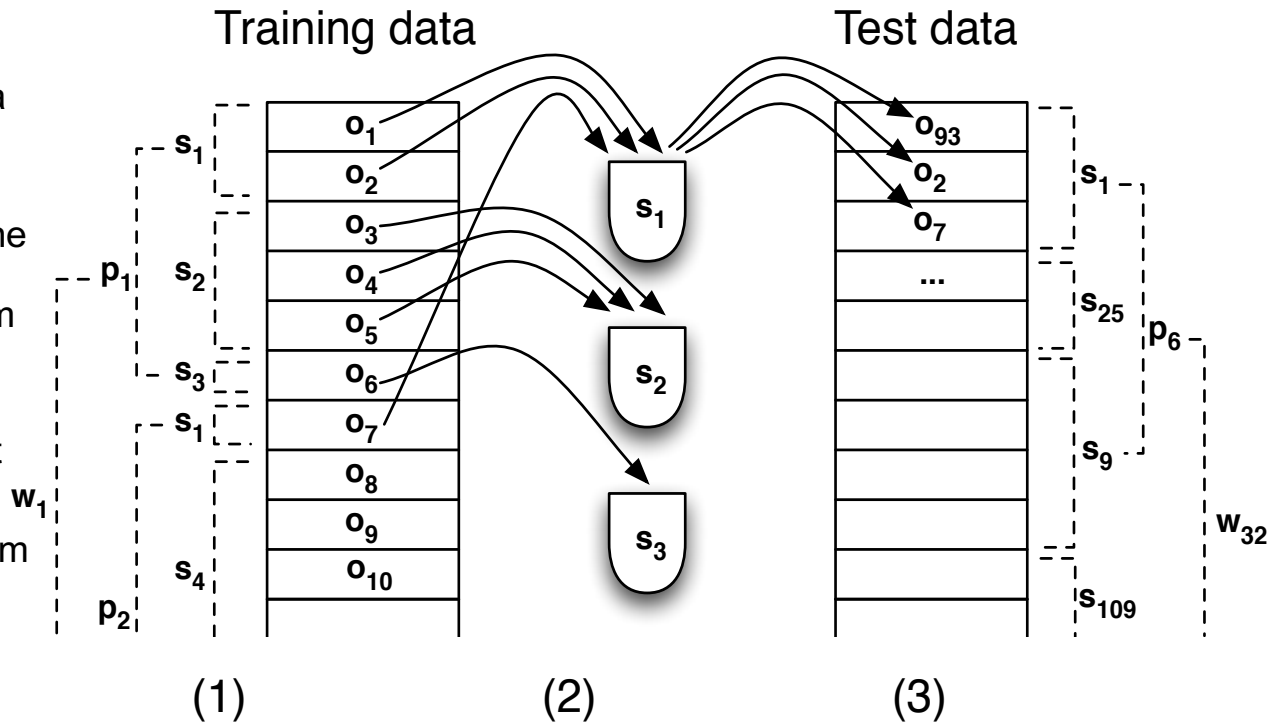
- After normally trained an acoustic model, we have
  - Transition probabilities
  - Emission probabilities
  - Original real data
  - Original transcript
  - Pronunciation dictionary

+ Pseudo speech data

- This reconstructed pseudo data has exactly the same length in frames with exactly the same state sequence and alignment.
- Each frame is generated/picked-up from only one of mutually independent states, based on independent multivariate distributions.
- That is, this data will completely satisfy the suspicious assumptions, except for that resampled data ignores the diagonal normal output distribution.

Frame level resampling

- Think of one “urn” for each state that holds observations
- Put all observations from the training data that are in that state into the urn
- Create new test utterances by Creating the same state sequence and selecting observations for each state randomly from the urn
- If the observations are really independent then it shouldn’t matter
  - what instance of a state they came from
  - what order they are in within the state
  - Which speaker they are from

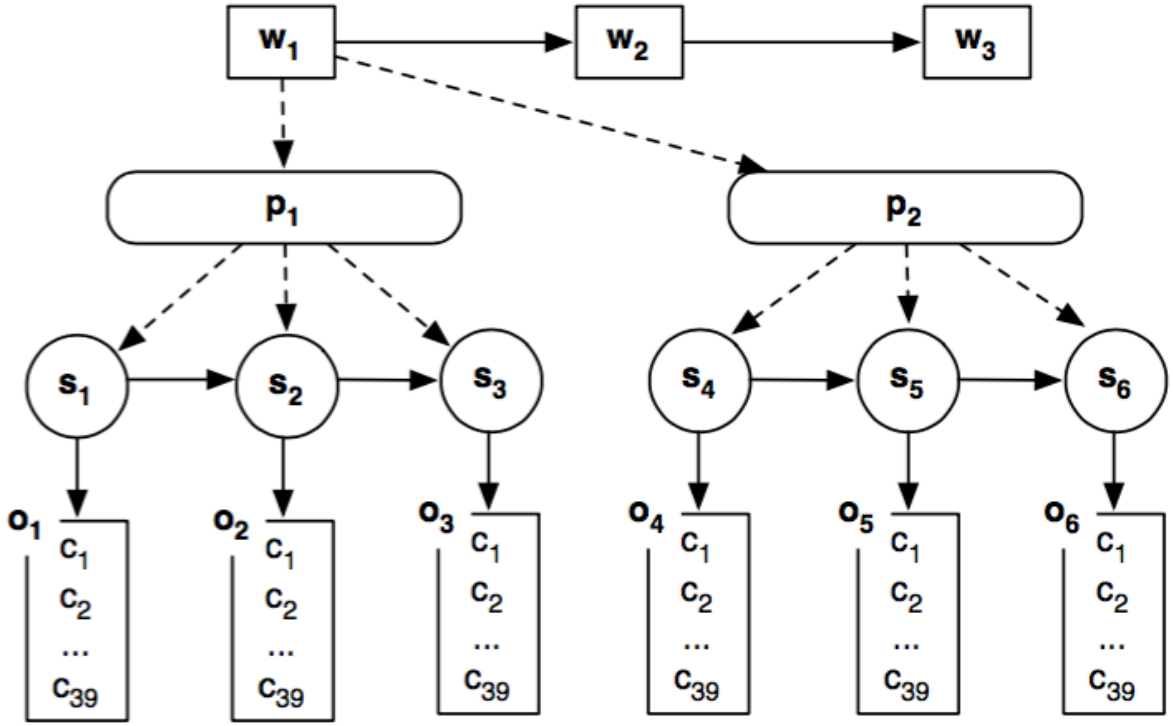


+ Results from Wegmann et al 2010

Dataset	WER
Original REAL speech data	.18
simulated	.02
resampled	.05
simulated using full cov matrix	.03

Conclusion: We have a serious problem in our model assumptions, and diagonal simplification is definitely not the problem.

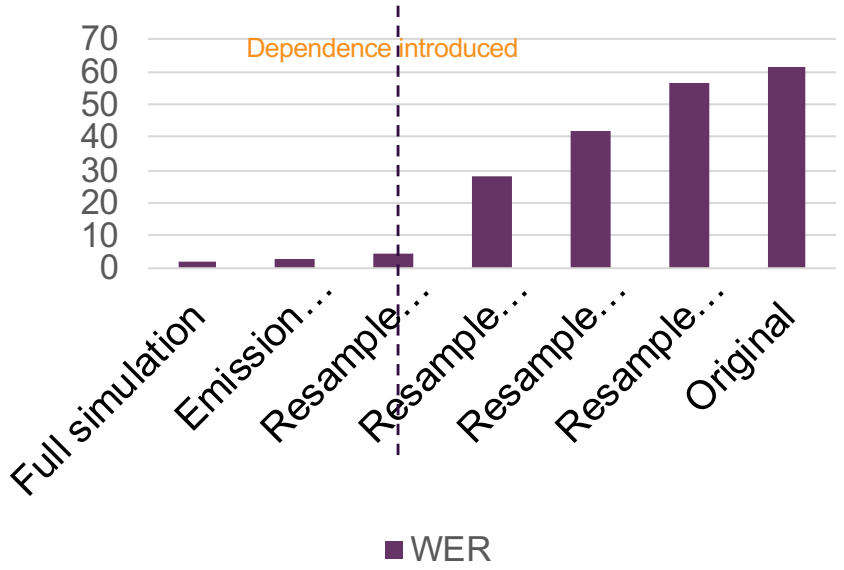
Multi-level resampling (Gillick et al 2011)



- Same idea, similar procedure but on
  - state level
  - phone level
  - word level

Results on SWBD from Gillick et al 2011

Dataset	WER
Original	61.5
Full simulation	2.4
Emission only simulation	3.0
Resample frames	4.5
Resample states	28.2
Resample phonemes	42.1
Resample words	56.4



Conclusion: the largest increase in WER is observed when we move from frame resampling to state resampling ➔ this is where we first need to look at!

How can we fix this? - Some suggestions from Morgan et al 2013

- Diagnose, diagnose, diagnose.
  - We need diagnostic analysis.
  - Not simply seeing WER/perplexity going down, we need some kind of methodology of specificity and efficiency.
  - Encouraging a diagnostic spirit could have very broad effects.