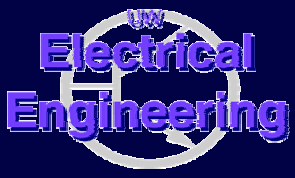


Graphical Model Research in Audio, Speech, and Language Processing

Jeff A. Bilmes

University of Washington

Department of EE, SSLI-Lab

Acknowledgements

- Thanks to the following people:
  - Chris Bartels – University of Washington
  - Ozgur Cetin – University of Washington
  - Karim Filali – University of Washington
  - Katrin Kirchhoff – University of Washington
  - Karen Livescu – MIT
  - Brian Lucena – University of Washington
  - Thomas Richardson – University of Washington
  - Geoff Zweig – IBM
- Slides and list of references for this tutorial available: http://ssli.ee.washington.edu/~bilmes

Outline

I. Graphical Models Review

II. Speech Recognition Overview

III. Goals for GMs in Speech/Language

   A. Explicit control

   B. Latent Modeling (audio, speech, language)

   C. Observation Modeling

   D. Structure Learning

IV. Toolkits and Inference

Graphical Models (GMs)

- Graphical Models give us:

SALAD

- **Structure**
- **Algorithms**
- **Language**
- **Approximations**
- **Data-Bases**

Graphical Models (GMs)

GMs give us:

I. **Structure**: A method to explore the structure of "natural" phenomena (causal vs. correlated relations, properties of natural signals and scenes)

II. **Algorithms**: A set of algorithms that provide "efficient" probabilistic inference and statistical decision making

III. **Language**: A mathematically formal, abstract, visual language with which to efficiently discuss and intuit families of probabilistic models and their properties.

Graphical Models (GMs)

GMs give us (cont):

III. **Approximation**: Methods to explore systems of approximation and their implications.

I. Inferential approximation

II. Task dependent structural approximation

IV. **Data-base**: Provide a probabilistic "data-base" and corresponding "search algorithms" for making queries about properties in such model families.

Conditional Independence

- Notation: $X \perp\!\!\!\perp Y \mid Z \equiv$

$$P(x, y \mid z) = p(x \mid z)\, p(y \mid z)\ \forall \{x, y, z\}$$

- Many CI Properties (from Lauritzen 96)
  - $X \perp\!\!\!\perp Y \mid Z \Rightarrow Y \perp\!\!\!\perp X \mid Z$
  - $Y \perp\!\!\!\perp X \mid Z$ and $U=h(X) \Rightarrow Y \perp\!\!\!\perp U \mid Z$
  - $Y \perp\!\!\!\perp X \mid Z$ and $U=h(X) \Rightarrow X \perp\!\!\!\perp Y \mid \{Z,U\}$
  - $X_A \perp\!\!\!\perp Y_B \mid Z \Rightarrow X_{A'} \perp\!\!\!\perp Y_{B'} \mid Z$

    where A, B sets of integers, $A' \subseteq A$, $B' \subseteq B$

    $X_A = \{X_{A_1}, X_{A_2}, \ldots, X_{A_N}\}$

Directed GM (DGMs) (Bayesian Networks)

- When is $X_A \perp\!\!\!\perp X_B \mid X_C$?
- Only when *C* d-separates *A* from *B*, i.e. if:

  for all paths from A to B, there is a *v* on the path s.t. either one of the following holds:

  1. Either $\rightarrow v \rightarrow$ or $\leftarrow v \rightarrow$ and $v \in C$
  2. $\rightarrow v \leftarrow$ and neither *v* nor any descendants are in *C*
- Equivalent to "directed local Markov property" (CI of non-descendants given parents), plus others (again see Lauritzen '96)

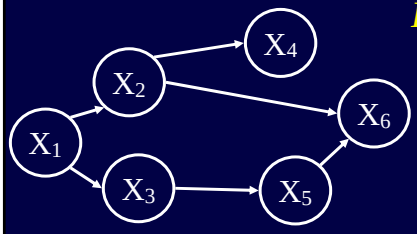
Conditional Independence

- Factorization (i.e., simplification)

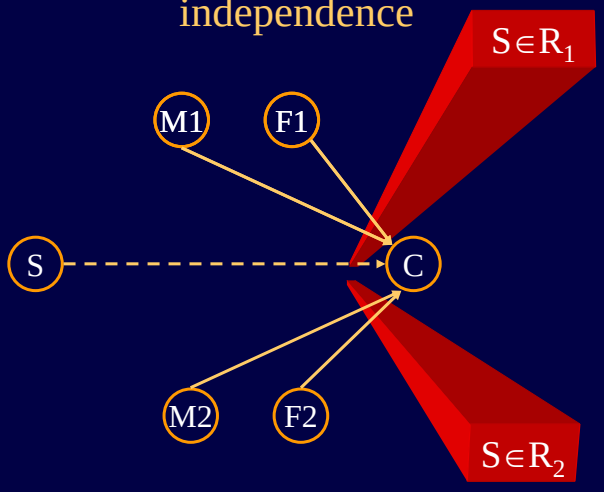
$$P(X_{1:T}) = \prod_i P(X_i \mid X_{1:i-1}) = \prod_i f(i^{th} \text{ subset of } X_{1:T})$$

- Better (more parsimonious) representations of the underlying problem since factors are local.
- Ex:

$$P(X_1, X_2, X_3, X_4, X_5, X_6) =$$
$$P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1)$$
$$P(X_4 \mid X_2)P(X_6 \mid X_2, X_5)P(X_5 \mid X_3)$$



Switching Parents: Value-specific conditional independence



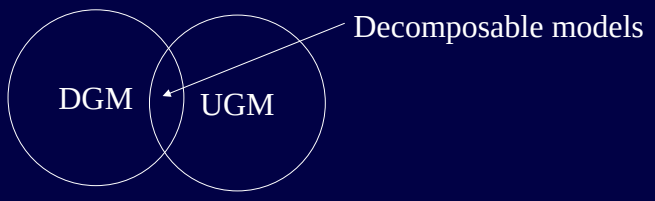
$$P(C \mid M1, F1, F2, F2) = \sum_i P(C \mid M_i, F_i, S = i)P(S \in R_i)$$

Undirected GMs

- When is $X_A \perp\!\!\!\perp X_B \mid X_C$?
- Only when *C* separates *A* from *B*. I.e., if:

  for all paths from A to B, there is a *v* on the path s.t. $v \in C$
- Simpler semantics than Bayesian networks.
- Equivalent to "global Markov property", plus others (again see Lauritzen '96)

Directed and Undirected models represent different families



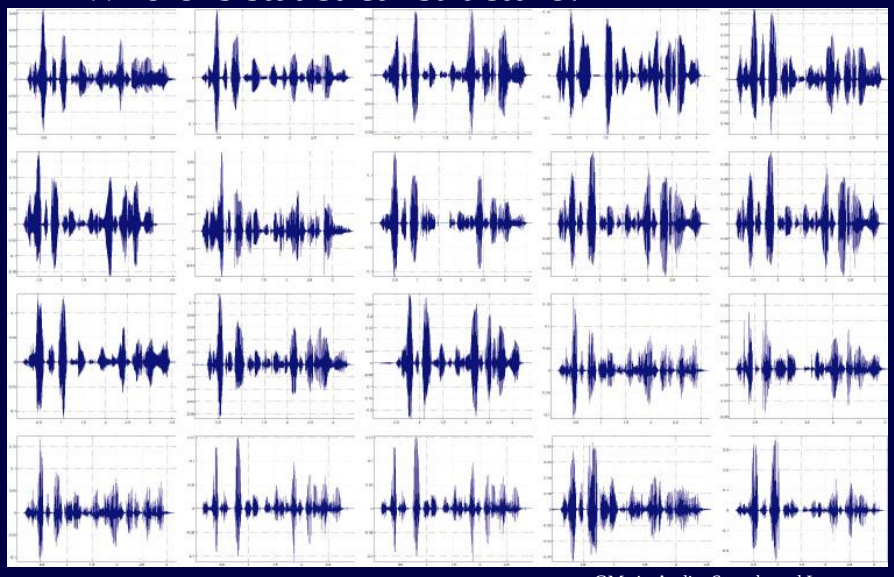
- The classic examples:

Why Graphical Models for Speech and Language Processing

- Expressive but concise way to describe properties of families of distributions
- Rapid movement from novel idea to implementation (with the right toolkit)
- GMs encompass many existing techniques used in speech and language processing but GM space is only barely covered
- Holds promise to replace the ubiquitous HMM
- Dynamic Bayesian networks and dynamic graphical models can represent important structure in "natural" time signals such as speech/language.

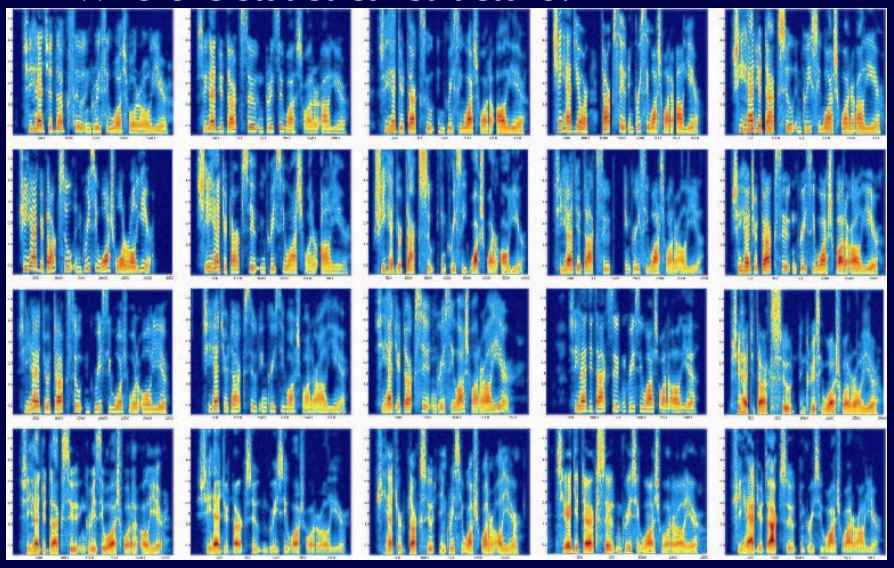
Time signals

- Where is statistical structure?



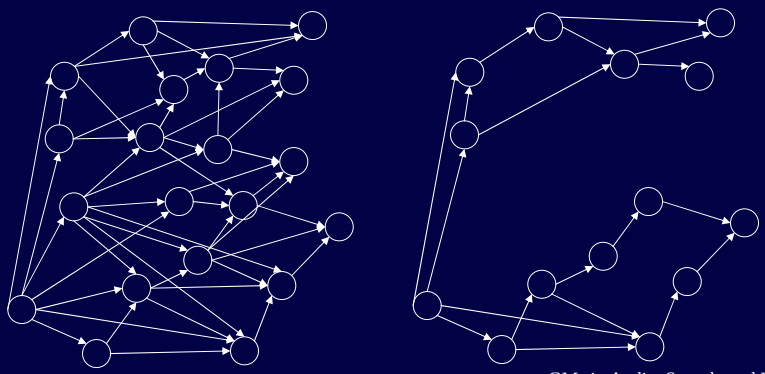
Spectrograms

- Where is statistical structure?

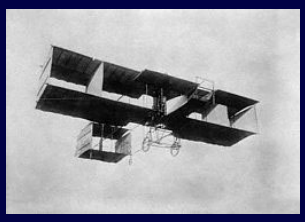


Structure of a Domain

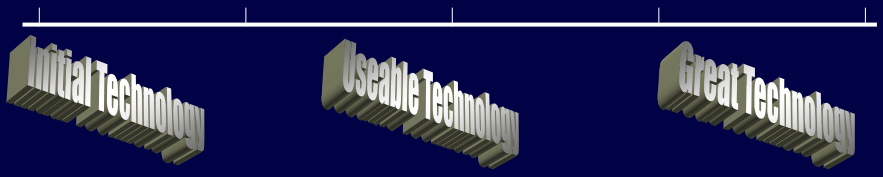
- Graphs represent properties of (auditory) objects from natural scenes.
- Goal: find minimal structure representing appropriate properties for a given task (e.g., object classification or ASR)



Speech Recognition

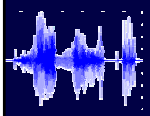


?



Research & Development needed to move to the right.

Automatic Speech Recognition: Broad View



Front End

$X_{1:T}$

Back End

Spoken word output hypotheses:
"How to recognize speech"
"How to wreck a nice beach"

Deterministic Signal Processing: Ex: Mel-frequency cepstral coefficients (MFCCs)

Length T sequence of feature vectors (so NxT matrix) T is often (but not always) known

Transform feature vectors into a string of words.

$P(W_{1:K} \mid X_{1:T})$

Ideal case, use Bayes decision rule

$$(K^*, W^*_{1:K}) = \operatorname*{argmax}_{K, W_{1:K}} \Pr(W_{1:K}, K \mid X_{1:T})$$

$$= \operatorname*{argmax}_{K, W_{1:K}} \Pr(X_{1:T} \mid W_{1:K}, K) \Pr(W_{1:K})$$

•Bayes Decision Theory (see Duda & Hart 73)

Generative vs. Discriminative Models

- Ideal case: discriminative model

$$P(W_{1:K} \mid X_{1:T})$$

- Too many classes for a discriminative model
  - 100k words, K = 10 ⇒ (100k)^10 classes
    - (not to mention we didn't consider all other K's)
- Generative model can help: $P(x_{1:T})$
- Use the "natural" hierarchy in speech/language:
  - Sentences are composed of words (W)
  - Words (W) are composed of phones (Q)
  - Phones (Q) are composed of Markov chain states (S)
  - States (S) are composed of acoustic feature vector sequences (X)
  - Acoustic feature vector sequences (X) are composed of noisy (e.g., channel distorted) versions thereof (Y)

Speech/Language Hierarchy
(time collapsed)

Word sequences

Phones – typically context-dependent, could be syllables, etc.

States – Markov chain states,

"Clean" Speech – the ideal speech signal without channel effects (additive & convolutional noise)

Speech – as received by a microphone or your ears, typically contains speech + unwanted material.

Other possible Q hidden variables

- Syllables – bigger than phones, smaller than words, perceptually meaningful unit
- Subphones (i.e., ½ or 1/3 of a phone)
- Context-dependent phones
  - Tri-phone (a phone in the context of its immediately preceding and following phone)
  - Better for temporal context and coarticulation
- Most common: 3-state tri-phones
  - Tri-phones that force the use of three states (S values).
  - Others are used, IBM uses 5 contextual phones on the left and right.
- Goal: not too many (curse of dimensionality, estimation problems) and not too few (accuracy).

Tri-phones Example

- $P(x|q_t)$ where q_t is a tri-phone
- x-y+z notation: phone y with
  - left context of x
  - right context of z
- Example transcription of "Beat it" : sil b iy t ih t sil
  - sil
  - sil-b+iy
  - b-iy-t (or b-iy+dx for Americans)
  - iy-t+ih (or iy-dx+ih)
  - t-ih+t (or dx-ih+t)
  - ih-t+sil
  - sil
- To further increase states: Word internal vs. cross-word tri-phones

A Generative Model of Speech

- Key goal, find a distribution over the variable-length set of feature vectors.

$$P(x_{1:T}|T) = \sum_{w_{1:K},K} P(x_{1:T}, w_{1:K}, K)$$

$$= \sum_{w_{1:T},K} \sum_{q_{1:M},M} P(x_{1:T}, q_{1:M}, M, w_{1:K}, K)$$

$$= \sum_{w_{1:T},K} \sum_{q_{1:M},M} \sum_{s_{1:T}} P(x_{1:T}, s_{1:T}, q_{1:M}, M, w_{1:K}, K)$$

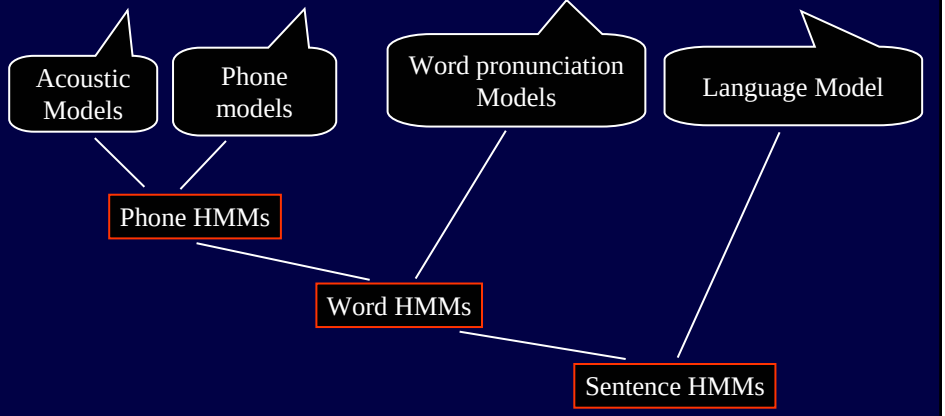
K words

M "phones"

T states

Generative Models of Speech

$$P(x_{1:T}, s_{1:T}, q_{1:M}, M, w_{1:K}, K) =$$
$$P(x_{1:T} \mid s_{1:T}) P(s_{1:T} \mid q_{1:M}, M) P(q_{1:M}, M \mid w_{1:K}, K) P(w_{1:K}, K)$$

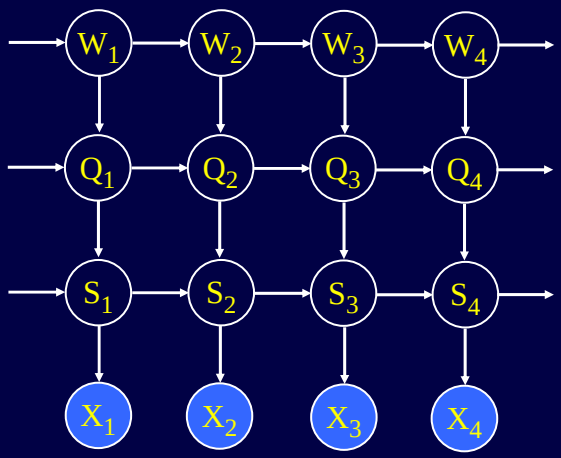


Solution implemented using search via dynamic programming

$$w^* = \max_{s,q,w} p(x, q, s, w)$$

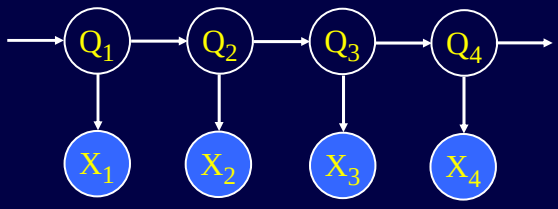
- Need optimized search algorithms
  - Viterbi decoding, time synchronous
  - stack decoding, A* search, time asynchronous
  - Both will heavily prune the search space, thus achieving a form of "approximate" inference

In other words, ASR has used Hierarchical HMMs for years, …



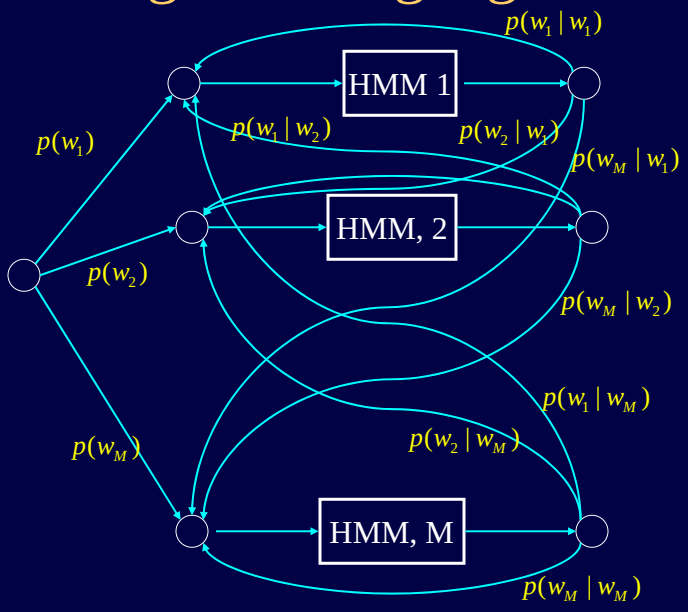
Hidden Markov Models

- … but, in existing speech systems, all of this complexity gets implicitly wrapped up (flattened) into an HMM

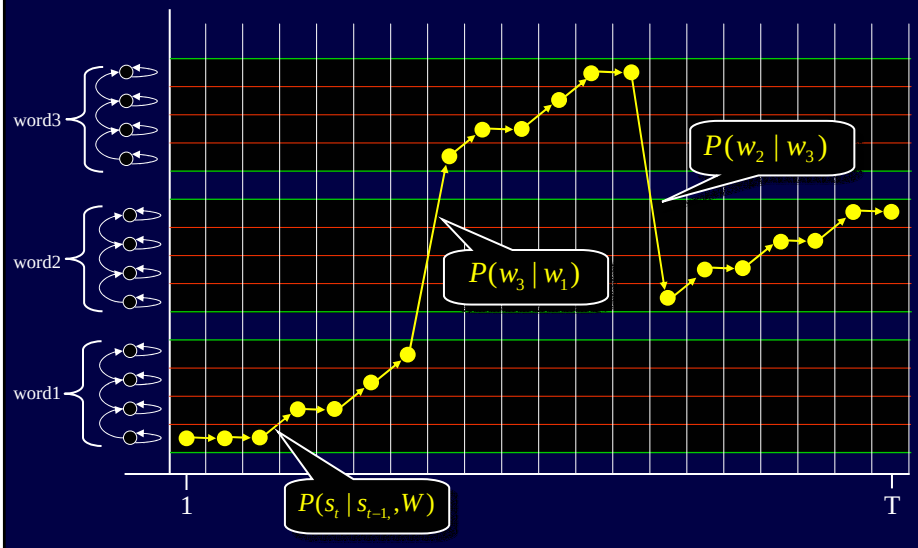


- Number of flattened states is strongly dependent on language model
  - Bi-gram language model: $P(w_i \mid w_{i-1,})$
  - Tri-gram language model: $P(w_i \mid w_{i-1,} w_{i-2})$

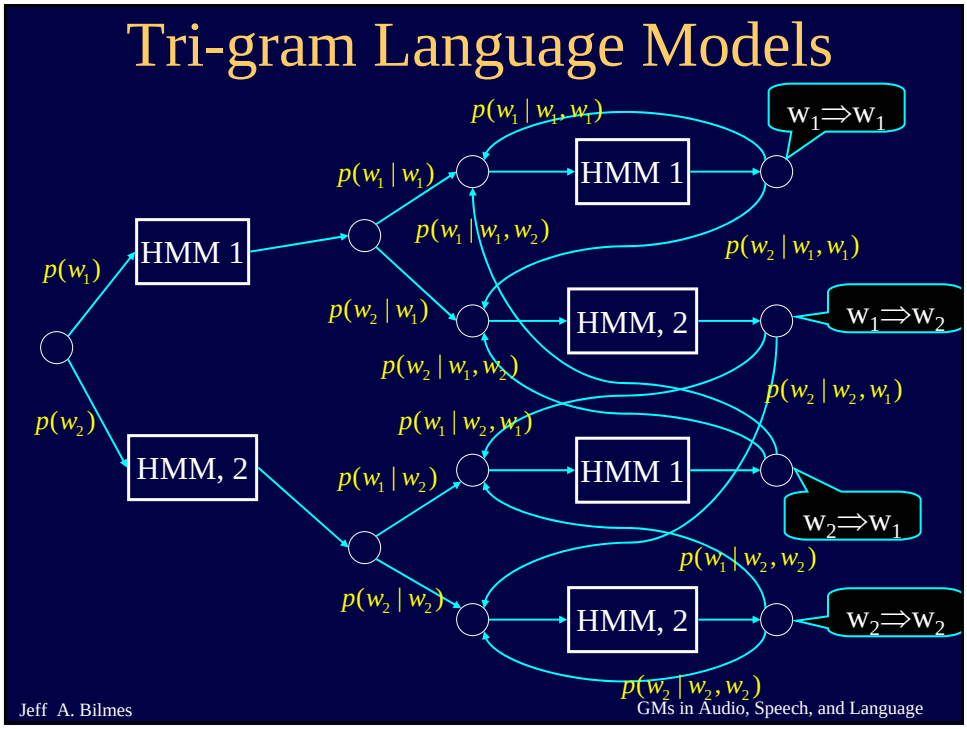
Bi-gram Language Models



HMM Lattice with bi-gram LMs

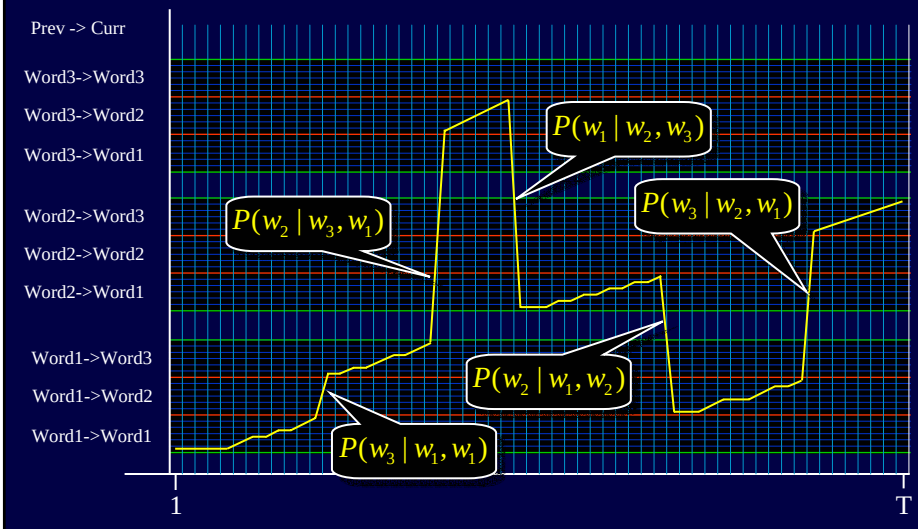


Tri-gram Language Models



HMM Lattice with tri-grams LMs

[word1],word1,word3,word2,word1,word2,word3

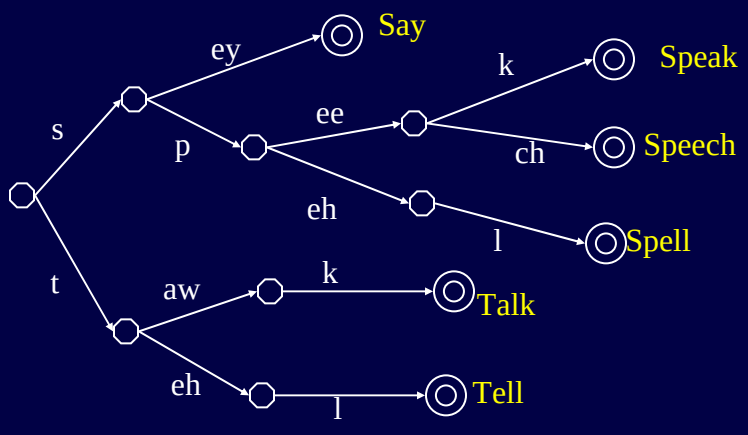


Challenges in Speech Recognition

- > 60k words, exhaustive examination of all words is infeasible since $|W|^2|Q||S|$ states.
- Even HMM decoding is a challenge
  - Clearly, large grid approach is infeasible
  - Pruning with a beam: try to discard unlikely partial hypotheses as soon as possible (without increasing error)
  - Explore word sequences in parallel (multiple partial hypotheses are considered at same time)

Tree-based Lexicons

- The speech/language hierarchy can help guide (and reduce computation in) the decoder



The Savior: Parameter Tying

- Generative model + speech/language hierarchy allows for massive amounts of parameter tying or sharing.
  - Same words in difference sentences or different parts of same sentence are the same
  - Same phones (subwords) in different words or in different parts of same word are the same
  - Certain states in different phones are merged
    - E.g., $p(x|S=i) = p(x|S=j)$ for the right i and j.
  - Certain observation parameters (e.g., means) are shared.
- Various ways to accomplish this:
  - backing off (like in language model)
    - [a-b+c] model backs off to [b+c] or to [a-b] etc.
  - Smoothing, interpolation, and mixing
  - clustering (widely used)

    Decision tree clustered tri-phones

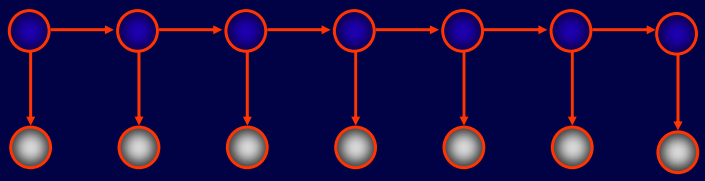
    both bottom up and top down clustering procedures.

Four Main Goals for GMs in Speech/Language

1. ***Explicit Control***: Derive graph structures that themselves *explicitly* represent control constructs
   - E.g., parameter tying/sharing, state sequencing, smoothing, mixing, backing off, etc.
2. ***Latent Modeling***: Use graphs to represent *latent information* in speech/language, not normally represented.
3. ***Observation Modeling***: represent structure over observations.
4. ***Structure learning***: Derive *structure* automatically, ideally to improve error rate while simultaneously minimizing computational cost.

Graph Control Structure Approaches

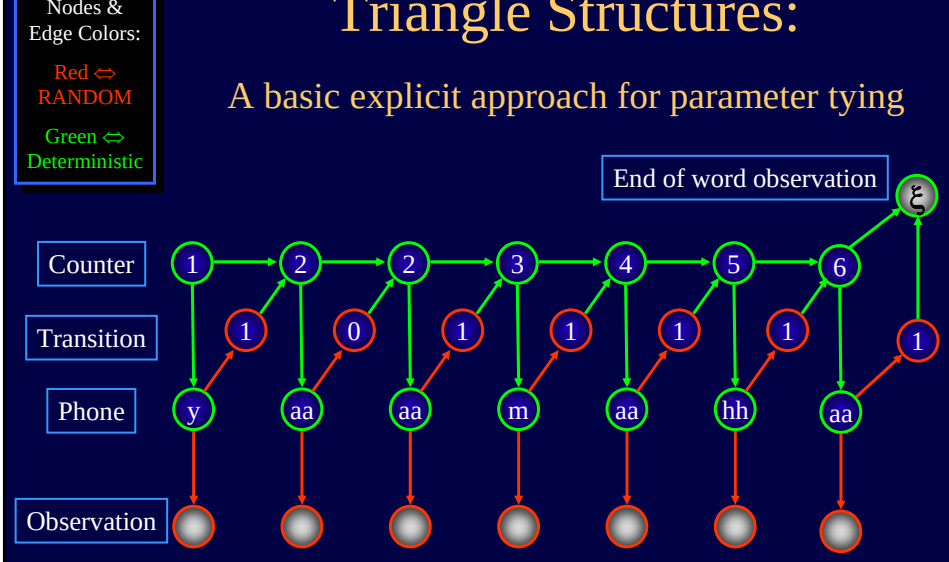
- The "*implicit*" graph structure approach
  - Implementation of dependencies determine sequencing through time-series model
  - Everything is flattened, all edge implementations are random



- The "*explicit*" graph structure approach
  - Graph structure itself represents control sequence mechanism and parameter tying in a statistical model.

Triangle Structures:

A basic explicit approach for parameter tying

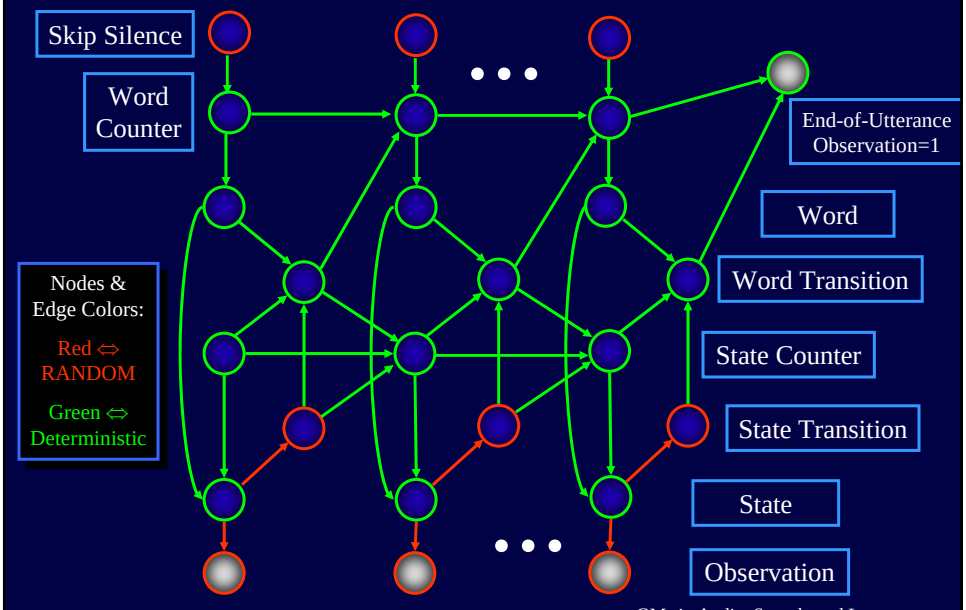


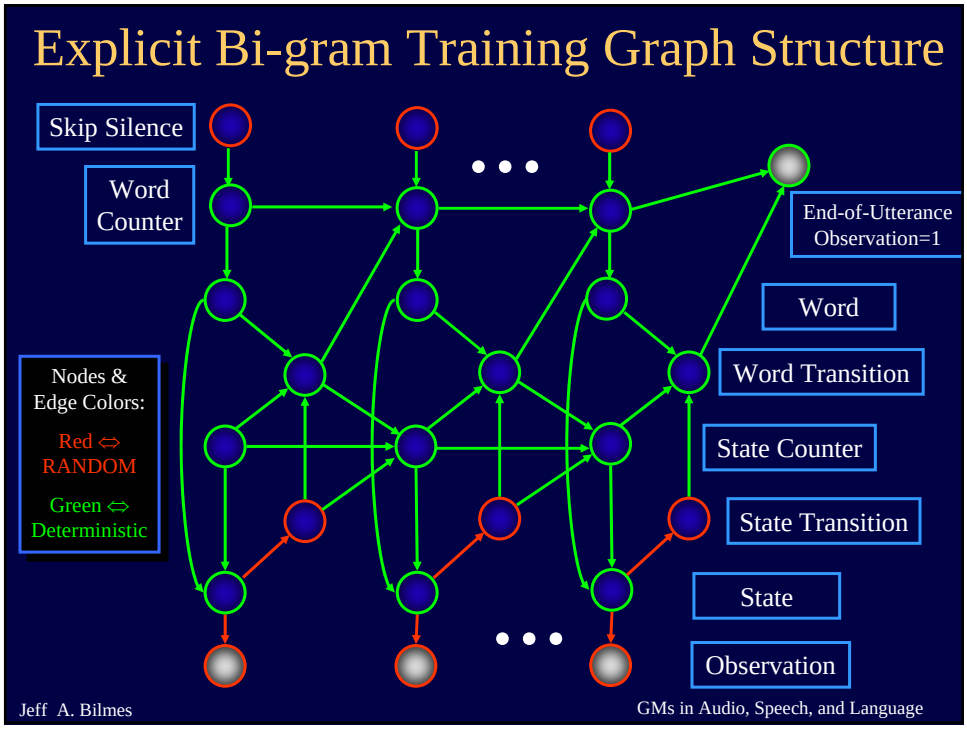
- Structure for the word "yamaha", note that /aa/ occurs in multiple places preceding different phones.

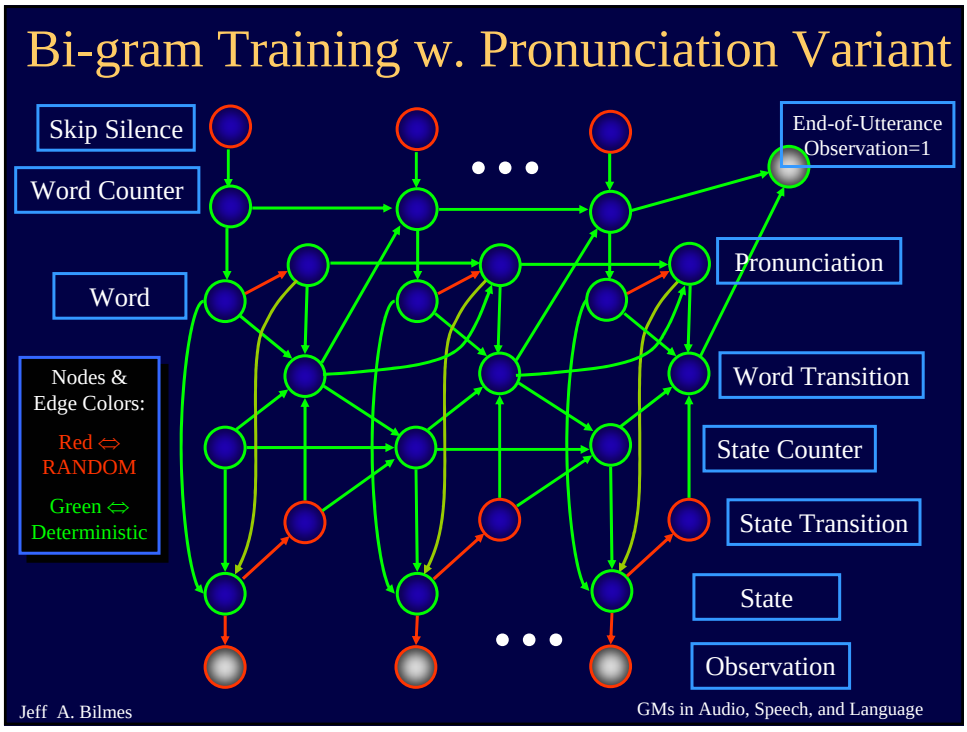
Key Points

- Graph explicitly represents parameter sharing
- Same phone at different parts of the word are the same: phone /aa/ in positions 2, 4, and 6 of the word “yamaha”
- Phone-dependent transition indicator variables yield geometric phone duration distributions for each phone
- Counter variable ensures /aa/’s at different positions move only to correct next phone
- Some edge implementations are deterministic (green) and others are random (red)
- End of word observation, gives zero probability to variable assignments corresponding to incomplete words.

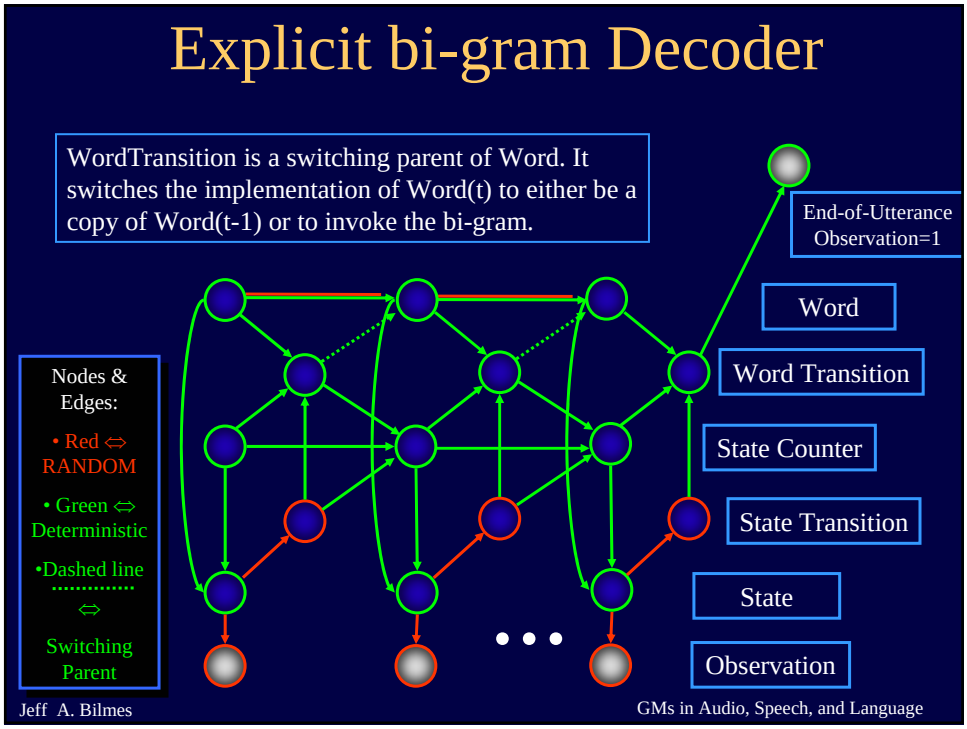
Explicit Bi-gram Training Graph Structure



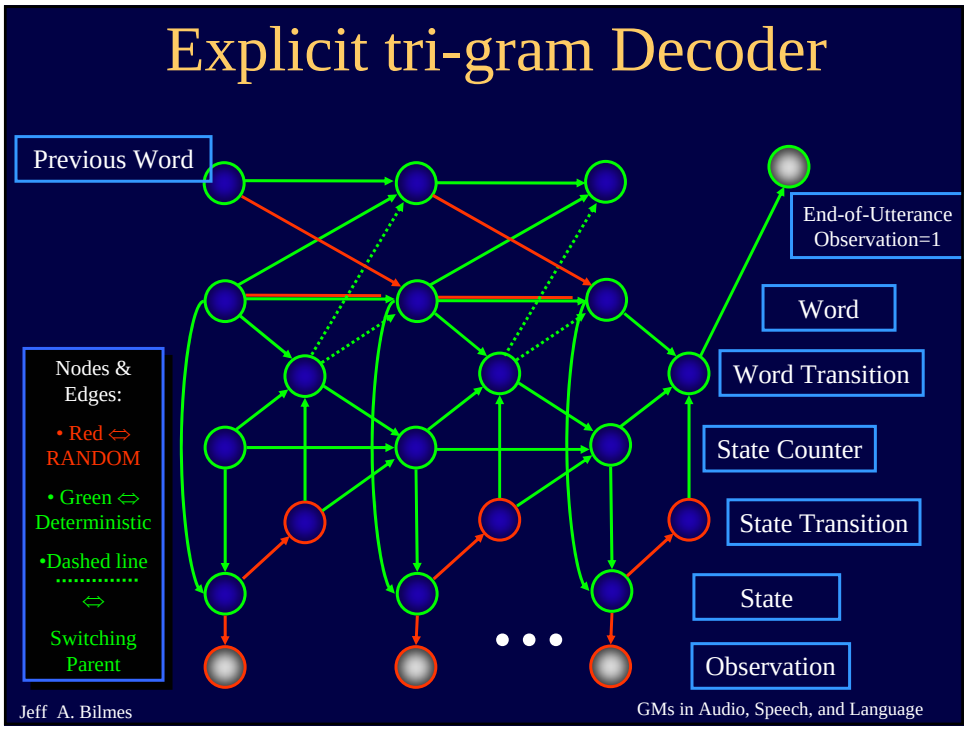

Explicit Bi-gram Training Graph Structure
Skip Silence
Word Counter
End-of-Utterance Observation=1
Word
Word Transition
State Counter
State Transition
State
Observation
Nodes & Edge Colors:
Red ⇔ RANDOM
Green ⇔ Deterministic
Jeff A. Bilmes
GMs in Audio, Speech, and Language


Bi-gram Training w. Pronunciation Variant
Skip Silence
Word Counter
End-of-Utterance Observation=1
Word
Pronunciation
Word Transition
State Counter
State Transition
State
Observation
Nodes & Edge Colors:
Red ⇔ RANDOM
Green ⇔ Deterministic
Jeff A. Bilmes
GMs in Audio, Speech, and Language

Explicit bi-gram Decoder



Explicit tri-gram Decoder

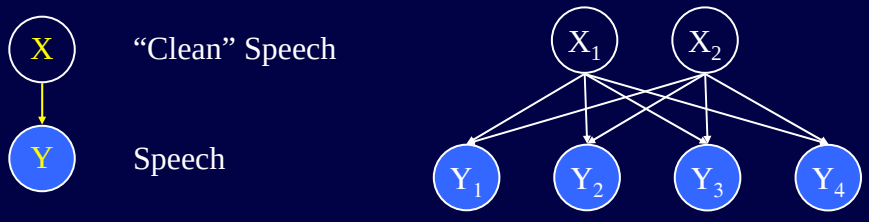


From Explicit Control to Latent Modeling

1. So far, each graph has been essentially still an HMM (in disguise).
2. Most edges were deterministic.
3. In latent modeling, we move more towards representing and learning additional information in (factored) hidden space.
4. Factored representations place constraints on what would be flattened HMM transition matrix parameters thereby potentially improving estimation quality.

Graphs for Speech/Audio Transformations

(feature level)

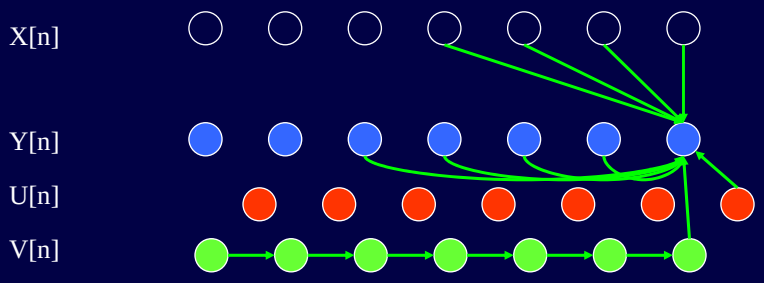


- The data $Y_{1:4}$ is explained by the two (marginally) independent latent causes $X_{1:4}$
- Many techniques used:
  - Principle components analysis
  - Factor Analysis
  - Independent Component Analysis
  - Linear Discriminant Analysis (different graph than above)

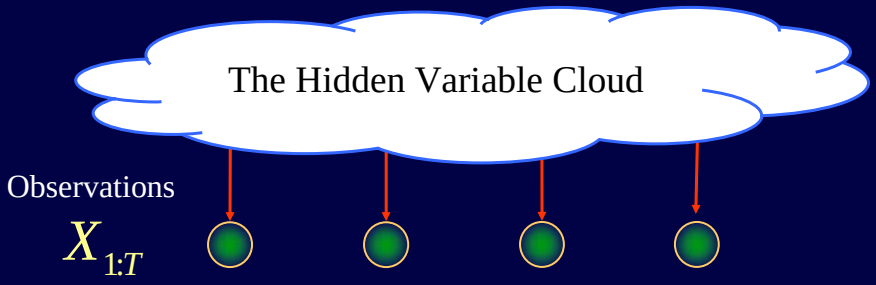
Speech/Audio De-Noising Models

(signal level)

- Latent variables X represent clean speech/sound at the signal/sample level for observed speech/sound Y
- Various forms of noise:
  - Convolutional: Moving Average, Auto-regressive
  - Independent Additive U
  - Structured Additive V

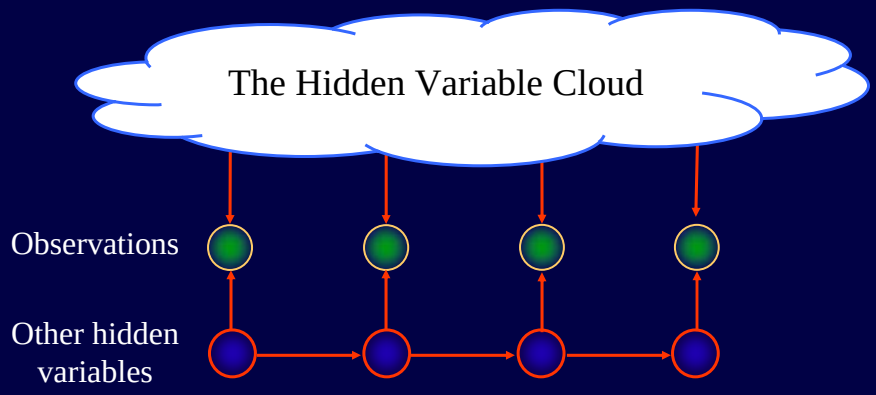


Latent Modeling



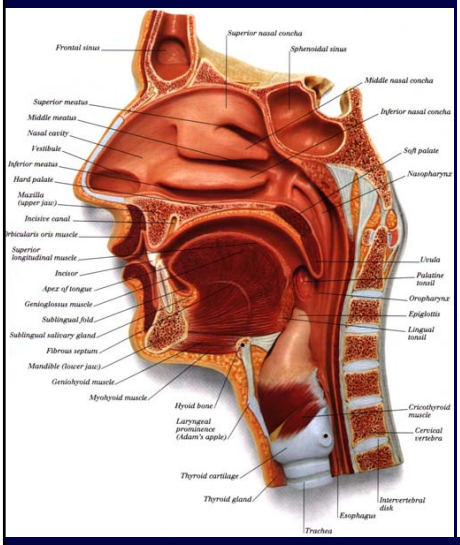
- Key Questions: What are the most important "causes" or latent explanations of the temporal evolution of the statistics of the vector observation sequence?
- How best can we factor these causes to improve parameter estimation, reduce computation, etc.?

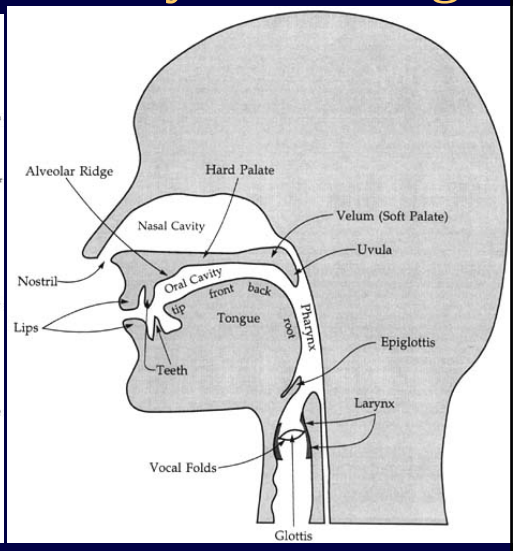
Latent X Modeling



- Where X = gender, speaker cluster, speaking rate, noise condition, accent, dialect, pitch, formant frequencies, vocal tract length, etc.
- We elaborate upon latent articulatory modeling…

Ex: Latent Articulatory Modeling

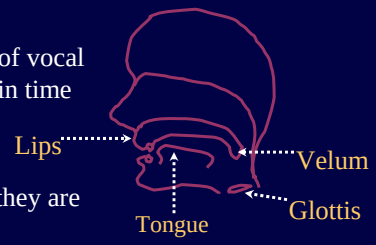




Pictures from Linguistics 001, University of Pennsylvania

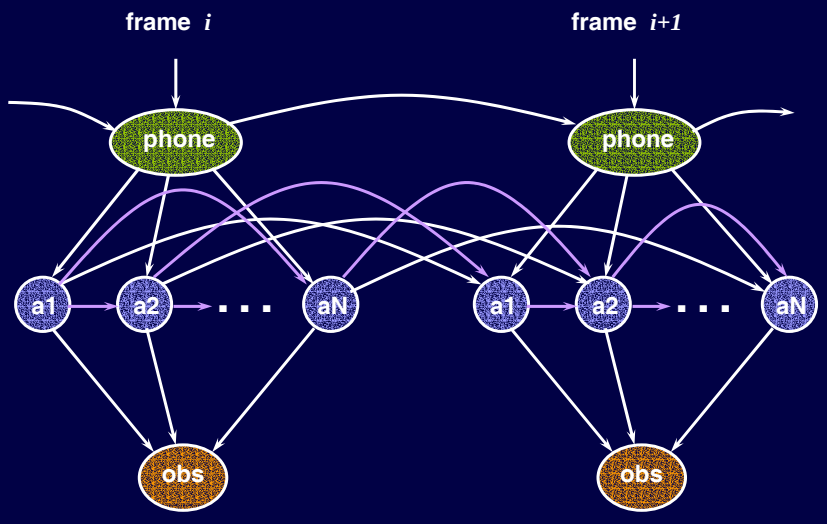
Latent Articulatory modeling.

- Definition:
  - Articulatory features specify complete state of vocal tract (directly or implicitly) at a given point in time
  - Features can be binary/multivalued, discrete/continuous, partial/complete
  - They are one of the "causes" of speech, yet they are hidden when given only acoustic signal



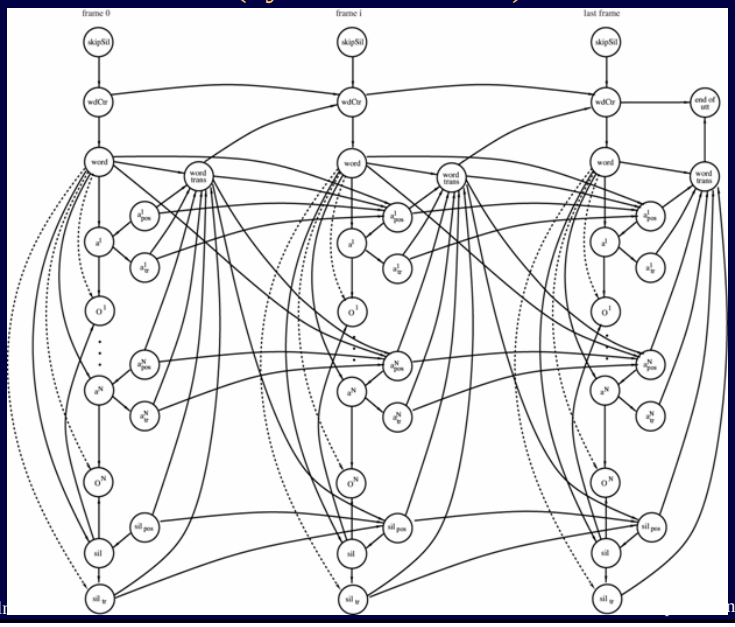
- Motivation:
  - Speech described by asynchronous articulator motion rather than by phones with synchronous start and end times
  - Articulatory features concisely represent coarticulatory effects such as nasalization and inserted stop closures ("warmth" → "warmpth")
  - Articulatory features can help recover information; e.g., a vowel is more likely to be nasalized if following nasal is deleted
  - Pronunciation modeling: phones are more likely to be modified by articulatory change rather than replaced with entirely different phones

Phone-based Articulatory graphical model



Phone-free Articulatory Graph

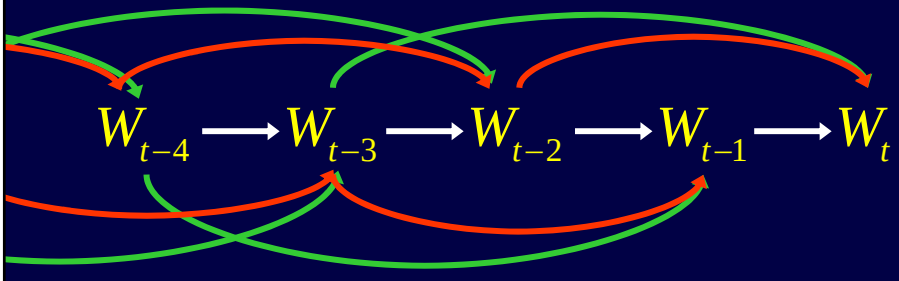
(by Karen Livescu)



Standard Language Modeling

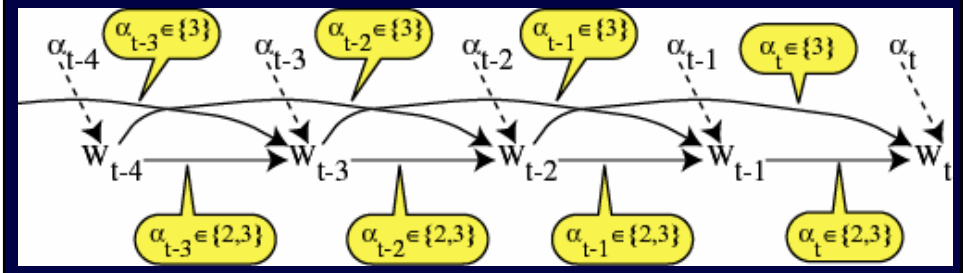
- Example: standard tri-gram

$$P(w_t \mid h_t) = P(w_t \mid w_{t-1}, w_{t-2}, w_{t-3})$$



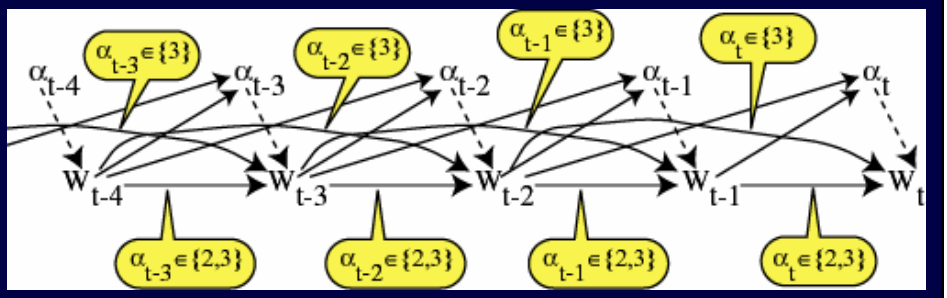
Interpolated Uni-,Bi-,Tri-Grams

$$P(w_t \mid h_t) = P(\alpha_t = 1)P(w_t) + P(\alpha_t = 2)P(w_t \mid w_{t-1}) + P(\alpha_t = 3)\,p(w_t \mid w_{t-1}, w_{t-2})$$



Conditional mixture tri-gram

$$P(w_t \mid h_t) = P(\alpha_t = 1 \mid w_{t-1}, w_{t-2})P(w_t) + P(\alpha_t = 2 \mid w_{t-1}, w_{t-2})P(w_t \mid w_{t-1}) + P(\alpha_t = 3 \mid w_{t-1}, w_{t-2})\,p(w_t \mid w_{t-1}, w_{t-2})$$



Skip Bi-gram

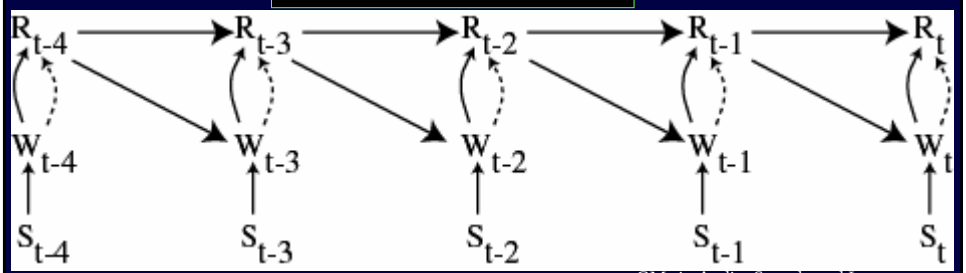
- Often there is silence between words
  - "fool me once <sil> shame on <sil> shame on you"
- Silence might not be good predictor of next word
- But silence lexemes should be represented since acoustics quite different during silence.
- Goal: allow silence between words, but retain true word predictability skipping silence regions.
- Switching parents can facilitate such a model.

Skip Bi-gram

$$p(r_t \mid w_t, r_{t-1}) = \begin{cases} \delta_{r_t = r_{t-1}} & \text{if} \quad w_t = sil \\ \delta_{r_t = w_t} & \text{if} \quad w_t \neq sil \end{cases}$$

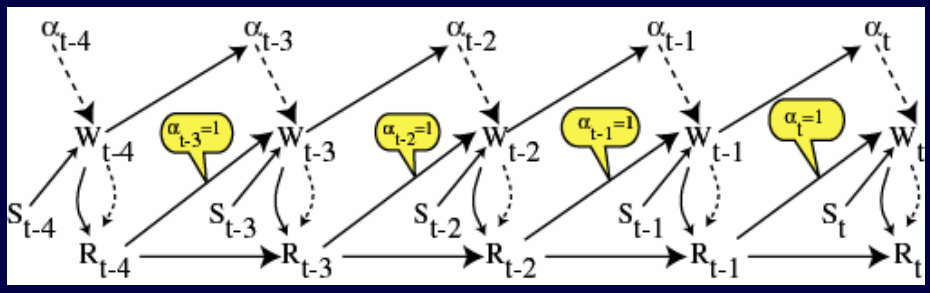
$$p(w_t \mid s_t, r_{t-1}) = \begin{cases} \delta_{w_t = sil} & \text{if} \quad s_t = 1 \\ p_{bigram}(w_t \mid r_{t-1}) & \text{if} \quad s_t = 0 \end{cases}$$

$$p(s_t = 1) = \Pr(silence)$$



Skip bi-gram with conditional mixtures

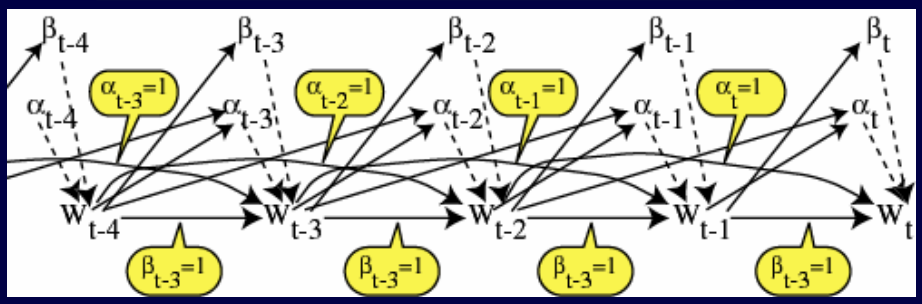
$$p_{bigram}(w_t \mid r_{t-1}) = P(\alpha_t = 1)P(w_t) + P(\alpha_t = 2)P(w_t \mid w_{t-1})$$



Two switching indicator variables

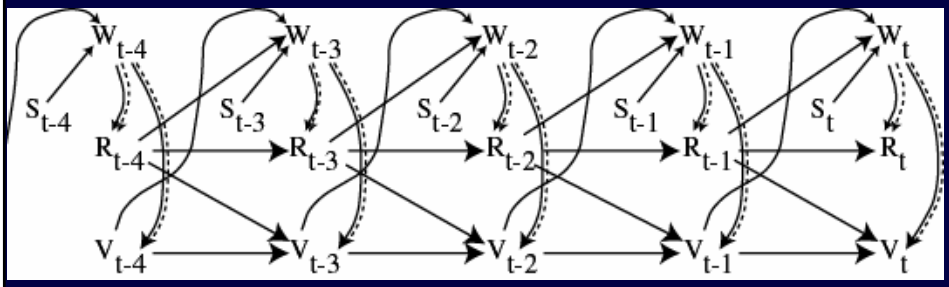
$$p(w_t \mid w_{t-1}, w_{t-2}) = P(\alpha_t = 1 \mid w_{t-1}, w_{t-2})P_{tri}(w_t \mid w_{t-1}, w_{t-2}) + P(\alpha_t = 0 \mid w_{t-1}, w_{t-2})P(w_t \mid w_{t-1})$$

$$p(w_t \mid w_{t-1}) = P(\beta_t = 1 \mid w_{t-1})P_{bi}(w_t \mid w_{t-1}) + P(\beta_t = 0 \mid w_{t-1})P(w_t)$$

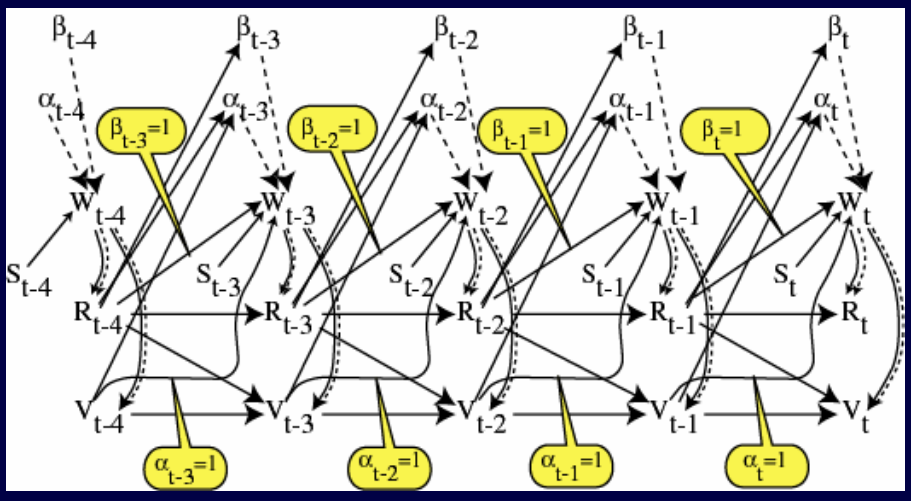


Skip trigram

- Similar to skip bi-gram, but skips over two previous <sil> tokens.
- P(City|<sil>,York,<sil>,New) = P(City|York,New)

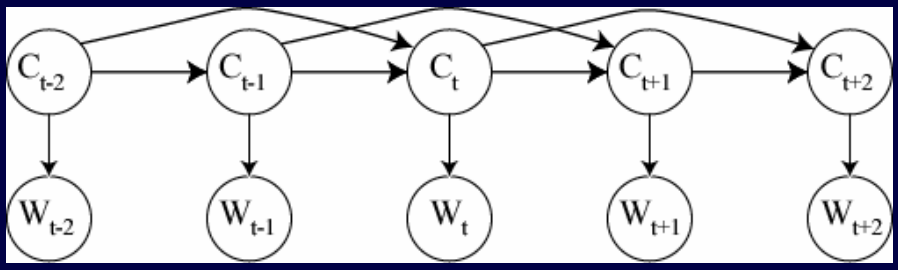


Putting it together: mixture and skip tri-gram.



Class Language Model

- When number of words large (>60k), can be better to represent clusters/classes of words
- Clusters can be grammatical or data-driven
- Just an HMM (perhaps higher-order)



Explicit Smoothing

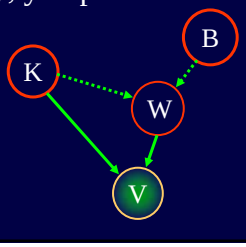
- Disjoint partition of vocabulary based on training-data counts: = {unk}∪ ∪
- = singletons, = "many-tons", unk=unknown
- ML distribution gives zero probability to unk.
- Goal: Directed GM that represents:

$$p(w) = \begin{cases} 0.5\,p_{ml}(\ ) & \text{if } w = unk \\ 0.5\,p_{ml}(w) & \text{if } w \in \\ p_{ml}(w) & \text{otherwise} \end{cases}$$

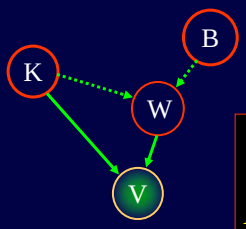
- Word variable is **like** a switching parent of itself (but of course can't be)

Explicit Smoothing

- Introduce two hidden variables K and B and one child observed variables V=1.
- Hidden variables are switching parents
  - K = indicator of singleton vs. "many-ton"
  - B = indicator of singleton vs. unknown word
- Observed child V induces "reverse causal" phenomena via its dependency implementation
  - I.e., child says "if you want me to give you non-zero probability, you parents had better do X"



Explicit Smoothing



$$P(B=1) = P(B=0) = 0.5$$

$$P(K=1) = 1 - P(K=0) = P(\ \ )$$

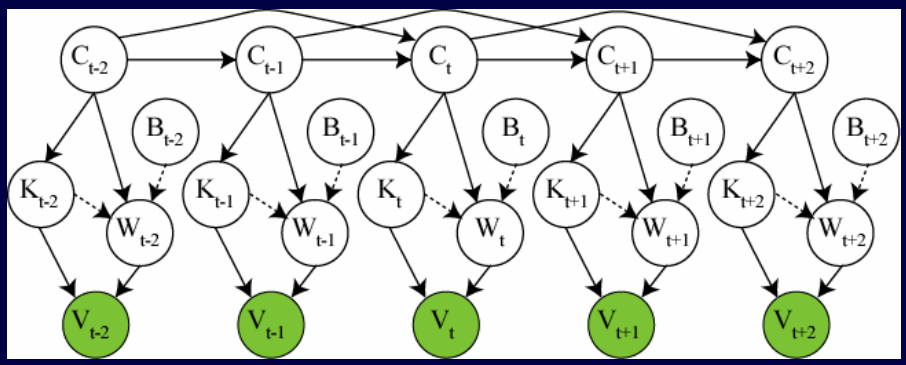
$$p(w \mid k,b) = \begin{cases} p_M(w) & \text{if } k=0 \\ p_S(w) & \text{if } k=1 \text{ and } b=1 \\ \delta_{w_t=unk} & \text{if } k=1 \text{ and } b=0 \end{cases}$$

$$P(V=1 \mid w,k) = 1\{(w \in \ \ , k=1) \text{ or } (w=unk, k=1) \text{ or } (w \in \ \ , k=0)\}$$

$$p_M(w) = \begin{cases} \frac{p_{ml}(w)}{p_{ml}(\ \ )} & \text{if } w \in \\ 0 & \text{else} \end{cases}$$

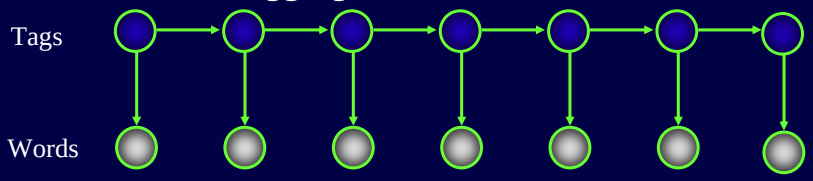
$$p_S(w) = \begin{cases} \frac{p_{ml}(w)}{p_{ml}(\ \ )} & \text{if } w \in \\ 0 & \text{else} \end{cases}$$

Putting it together: Class Language Model with smoothing constraints

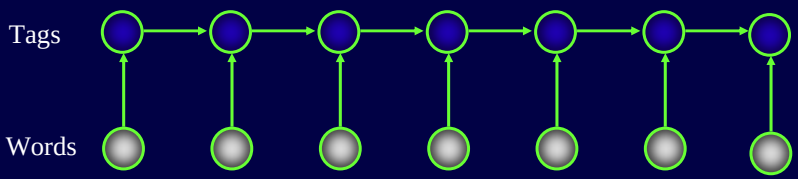


Part of speech Tagging

- Represent and find part-of-speech tags (noun, adjective, verb, etc.) for a string of words
- HMMs for word tagging



- Discriminative models for this task

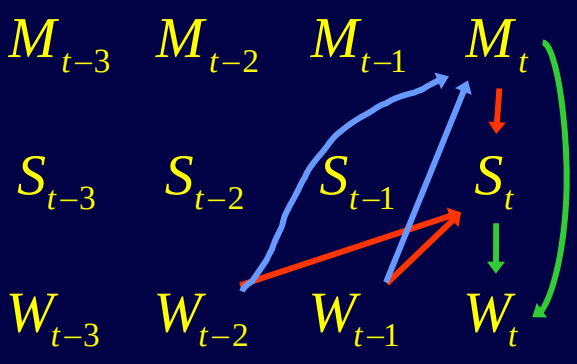


- Label bias issue and selection bias.

Factored Language Models

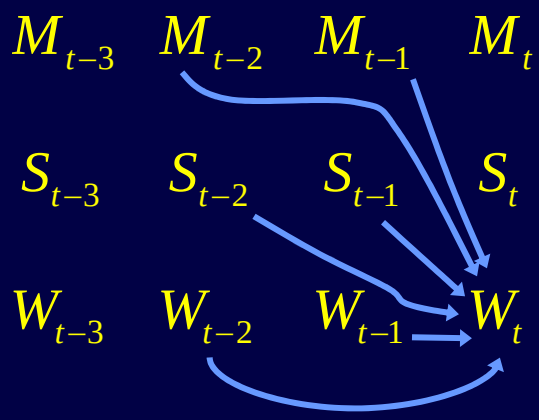
- Decompose words into smaller morphological or class-based units (e.g., morphological classes, stems, roots, patterns, or other automatically derived units).
- Produce probabilistic models over these units to attempt to improve language modeling accuracy and parameter estimation

Example with Words, Stems, and Morphological classes



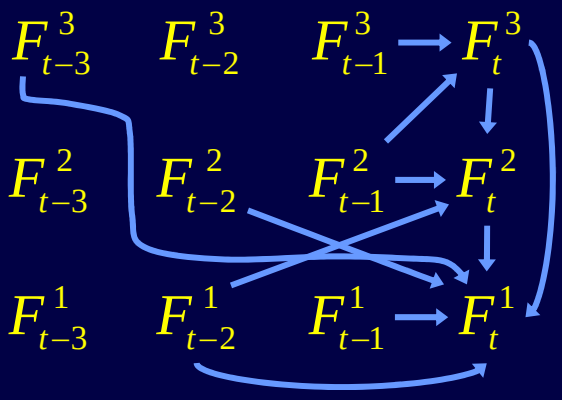
$P(w_t \mid s_t, m_t)$ $P(s_t \mid m_t, w_{t-1}, w_{t-2})$ $P(m_t \mid w_{t-1}, w_{t-2})$

Example with Words, Stems, and Morphological classes



$$P(w_t \mid w_{t-1}, w_{t-2}, s_{t-1}, s_{t-2}, m_{t-1}, m_{t-2})$$

In general



General Factored LM

- A word is equivalent to collection of factors.

$\{w_t\} \equiv \{f_t^{1:K}\}$ $\quad f^k$ = the kth factor

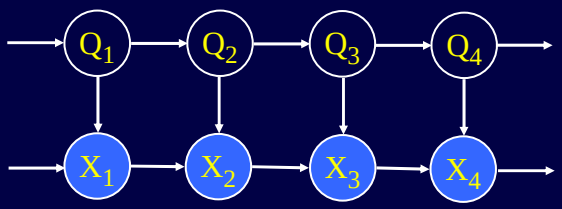
- E.g., if K=3

$$P(w_t \mid w_{t-1}, w_{t-2}) = P(f_t^1, f_t^2, f_t^3 \mid f_{t-1}^1, f_{t-1}^2, f_{t-1}^3, f_{t-2}^1, f_{t-2}^2, f_{t-2}^3)$$
$$= P(f_t^1 \mid f_t^2, f_t^3, f_{t-1}^1, f_{t-1}^2, f_{t-1}^3, f_{t-2}^1, f_{t-2}^2, f_{t-2}^3)$$
$$P(f_t^2 \mid f_t^3, f_{t-1}^1, f_{t-1}^2, f_{t-1}^3, f_{t-2}^1, f_{t-2}^2, f_{t-2}^3)$$
$$P(f_t^3 \mid f_{t-1}^1, f_{t-1}^2, f_{t-1}^3, f_{t-2}^1, f_{t-2}^2, f_{t-2}^3)$$

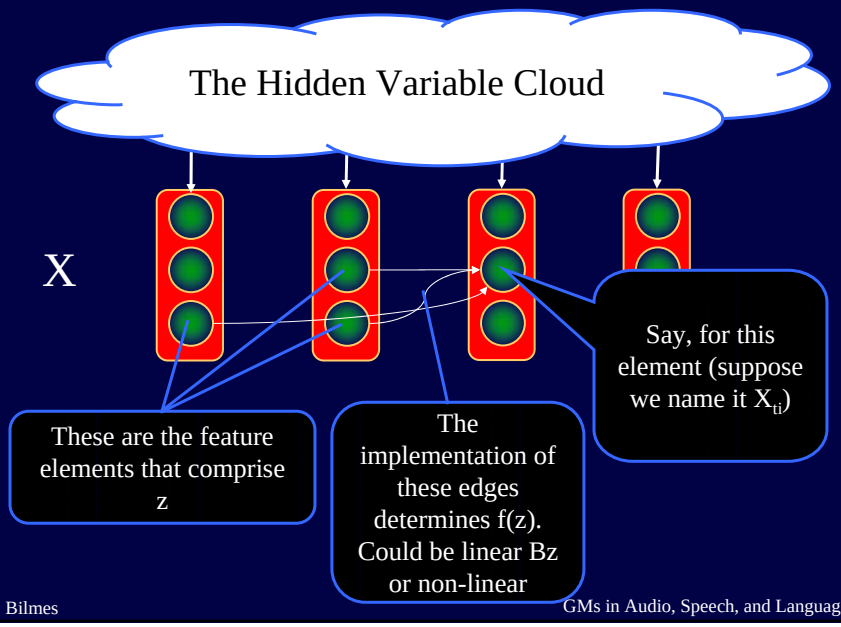
- Goal: find appropriate conditional independence statements to simplify while keeping perplexity and error low.
- structure learning problem

"Auto-regressive" HMMs

- Observation is no longer independent of other observations given current state
- Can not be represented by an HMM
- One of the first HMM extensions tried in speech recognition.

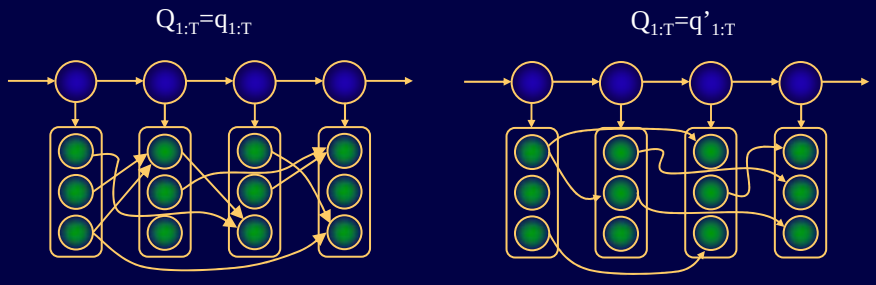


Observed Modeling

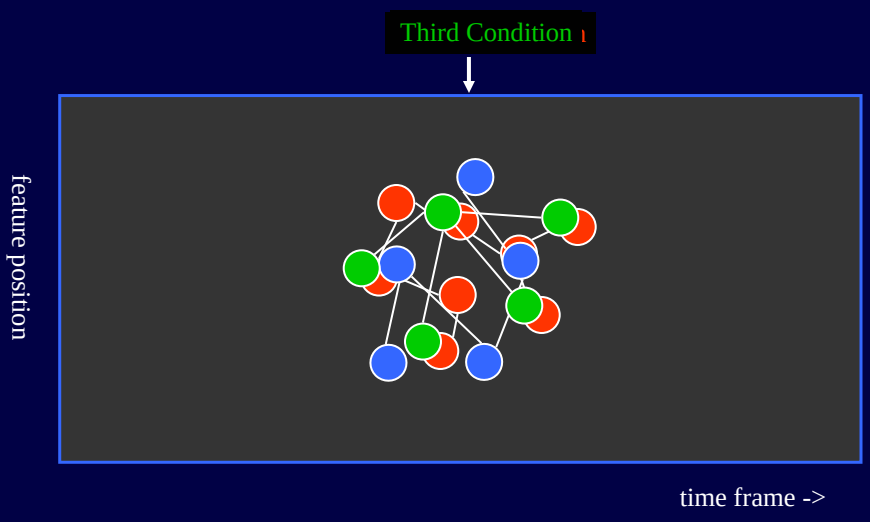


Buried Markov Models (BMMs)

- Markov chain is "further hidden" (buried) by specific element-wise cross-observation edges
- Switching dependencies between observation elements conditioned on the hidden chain.

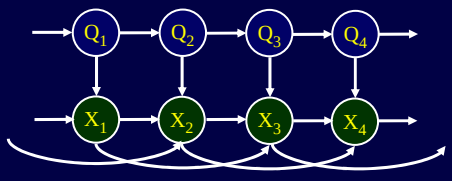


Switching Structure



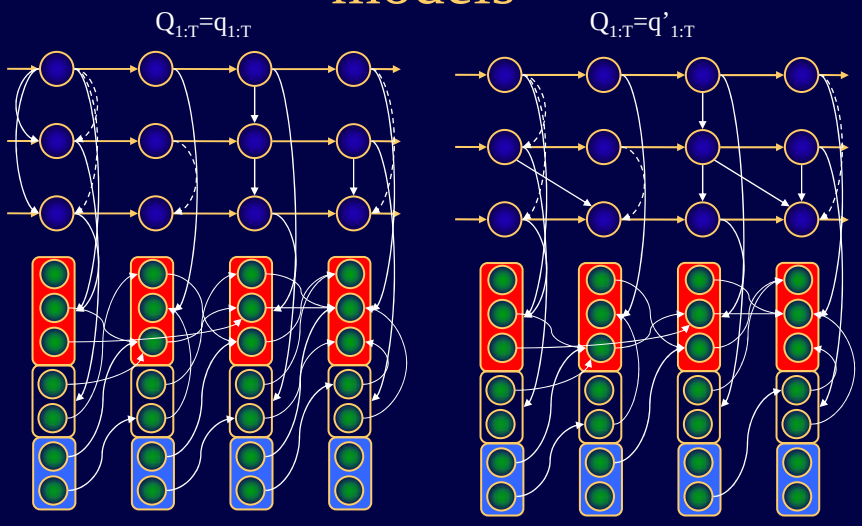
BMM Complexity

- AR-HMM(K)

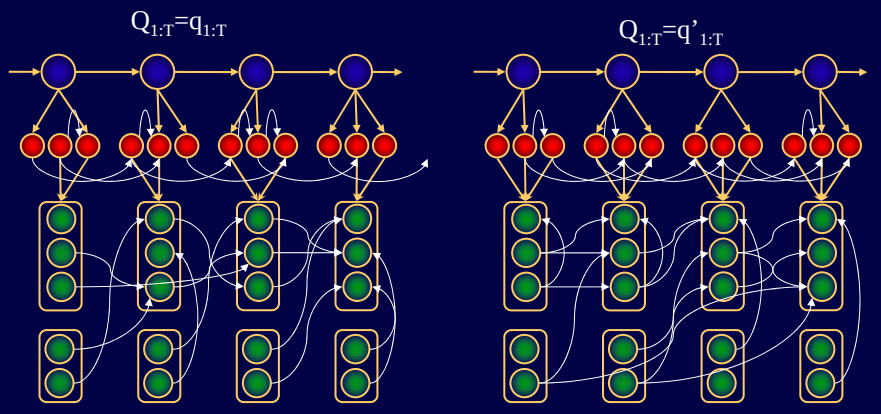


- Theorem: Triangulation comes for free after moralization in an AR-HMM.
- Theorem: Triangulated-by-moralization AR-HMM(K) has hidden clique state space size of at most 2.
- Therefore, BMMs have same asymptotic complexity as HMMs, but they can not be represented exactly via HMMs.

Multi-stream buried Markov models

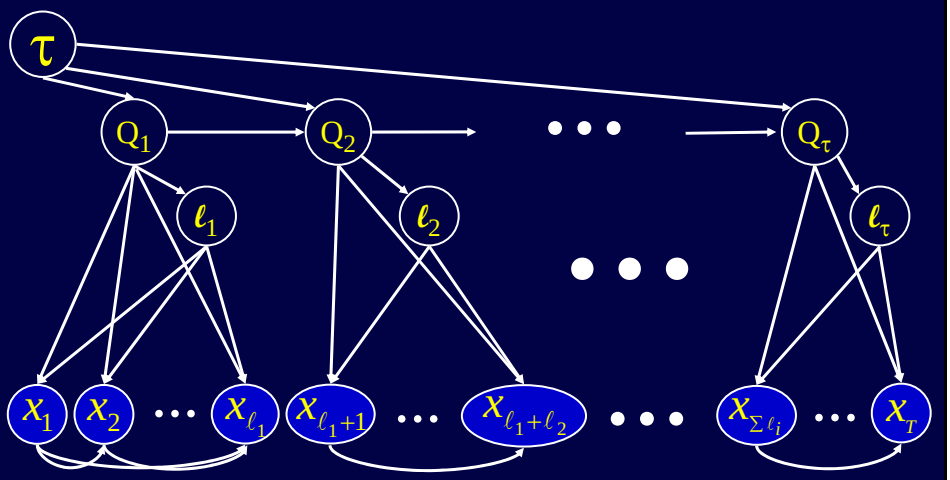


Buried Articulator Models (BAMs)



Segment Models as GMs

$$p(x_{1:T}) = \sum_{\tau} \sum_{q_{1:\tau}} \sum_{\ell_{1:\tau}} \prod_{i=1}^{\tau} p(x_{t(q_{1:\tau},\ell_{1:\tau},i,1)}, x_{t(q_{1:\tau},\ell_{1:\tau},i,2)}, \cdots, x_{t(q_{1:\tau},\ell_{1:\tau},i,\ell_i)} \mid q_i, \tau) p(q_i \mid q_{i-1}, \tau) p(\tau)$$



Discriminative structure learning

- Structure is typically learned to optimize marginal likelihood (e.g., statistical predictability)
- When the underlying goal is classification (regression), discriminative structure learning
- Structure is chosen to optimize conditional posterior of class variable (more generally, conditional likelihood) rather than marginal likelihood.
- Can still use generative models
- Structure edges can "switch" depending on current condition

Discriminative vs. Generative Models

- p(X|C) vs. p(C|X)
- Goals for recognition (classification) are different than for generative accuracy (e.g., synthesis)
- Generative models natural for ASR
- Approach: retain generative models but train discriminatively
  - Discriminative parameter training can occur for parameters of generative models also (e.g., maximum mutual information estimation on HMMs)
  - Discriminative structure learning.

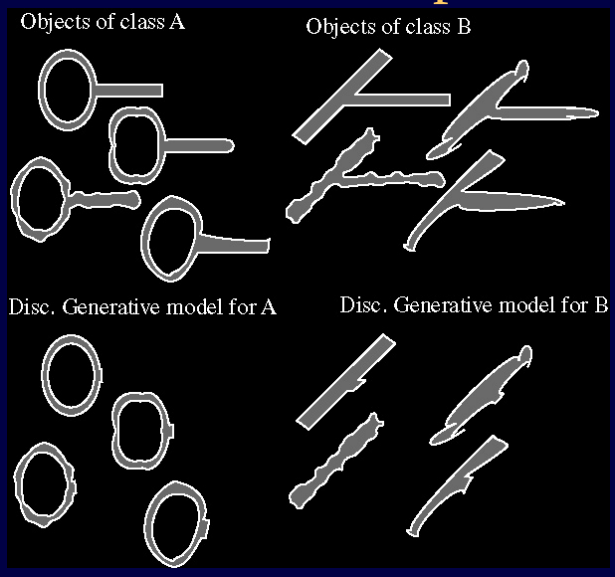
Discriminative Generative Models (DG models)

$$\mathcal{F} = \{ f(x:m) : \operatorname*{argmax}_m p(x \mid m)\, p(m) = \operatorname*{argmax}_m f(x,m)\, p(m) \}$$

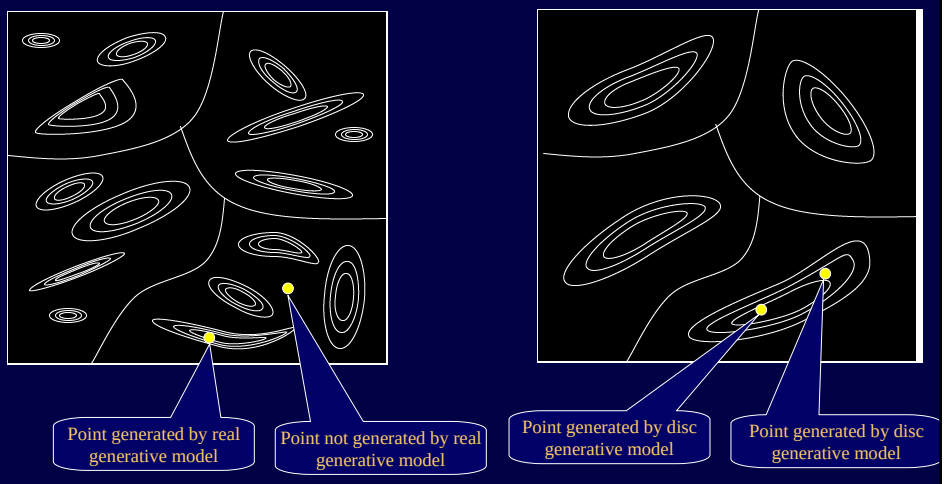
So choose the *f* that satisfies the above, but is as simple (few parameters, easy to compute) as possible.

Model might no longer "generates" samples accurately, but discriminates well.

Visual Example



Spatial Example

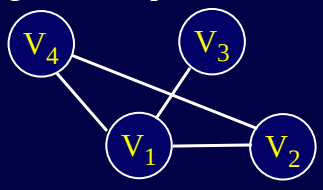


Structural Discriminability for Classification Performance

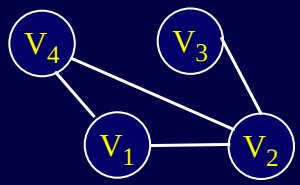
- Overall goal: model parsimony,
  - achieve same or better performance with same or fewer parameters.
  - Family of models that concentrate power only on what matters.
- Find structures that are inherently discriminative.
- Rough idea: represent the "unique" dependencies of each class, do not represent "common" dependencies for all classes that do not help classification.
- Models should be incapable of not discriminating
  - Distinct from being capable of discriminating

Structural Discriminability

Object generation process:

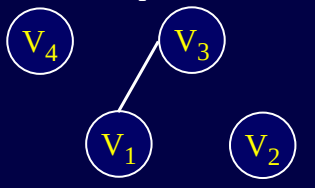


$P(V_1, V_2, V_3, V_4 \mid C = 1)$

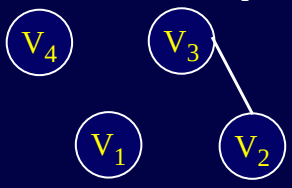


$P(V_1, V_2, V_3, V_4 \mid C = 2)$

Object recognition process: remove non-distinct dependencies that are found to be superfluous for classification, for example:



$\hat{P}(V_1, V_2, V_3, V_4 \mid C = 1)$



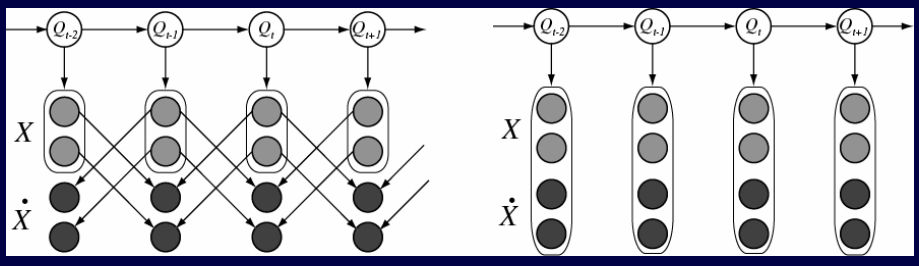
$\hat{P}(V_1, V_2, V_3, V_4 \mid C = 1)$

Comparison of generative vs. generative-discriminative models.

CASE	TYPE	WER	PARAMS
1	Generative	32.0%	207k
2	HMM	5.0%	157k
3	Discriminative	4.6%	157k

Why HMMs and delta features?

- Wrong can be right
  - HMM observations augment feature vector with temporal derivative estimates
  - Wrong generative structure is better from a discriminative point of view



The EAR Measure

$$I(X;Z \mid Q) - I(X;Z)$$

- EAR: Explaining Away Residual
- A way of approximating the discriminative quality of the edge between Z and X in context of Q.

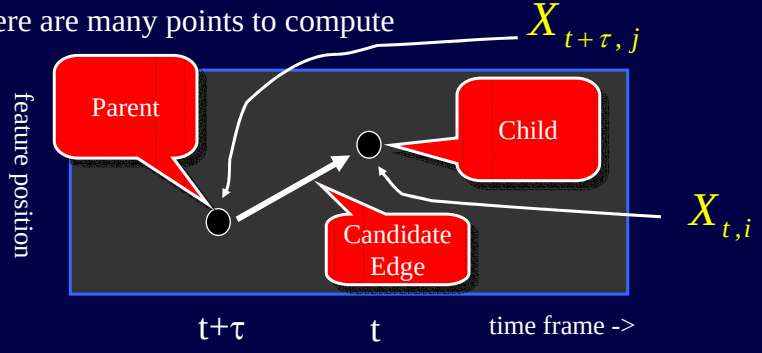
- Marginally Independent, Conditionally Dependent

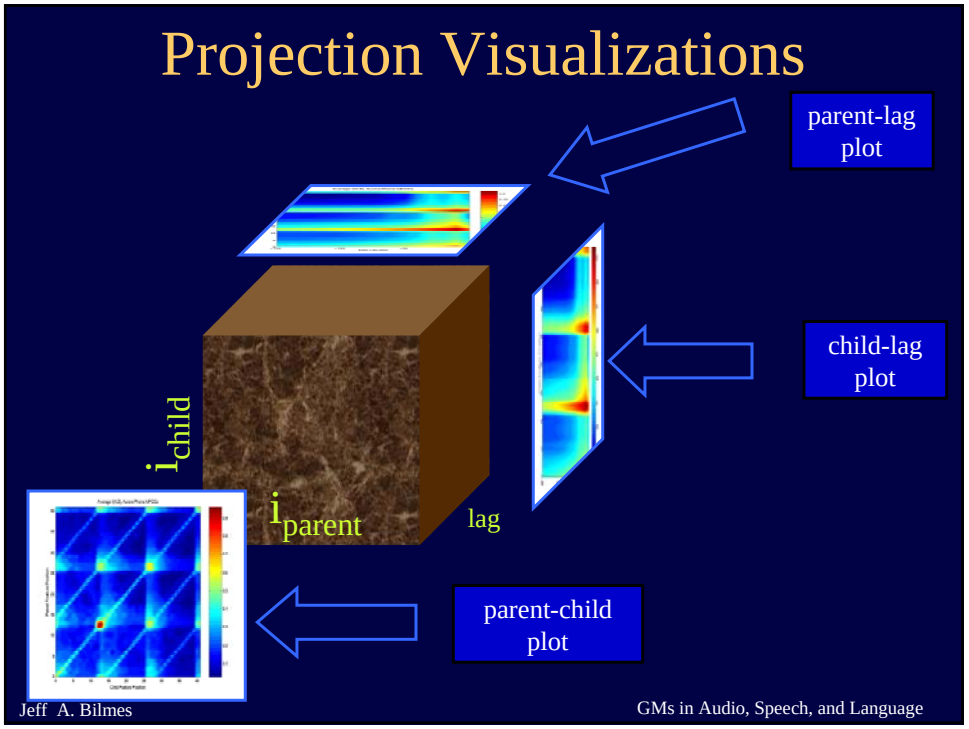
Visualizing Dependency

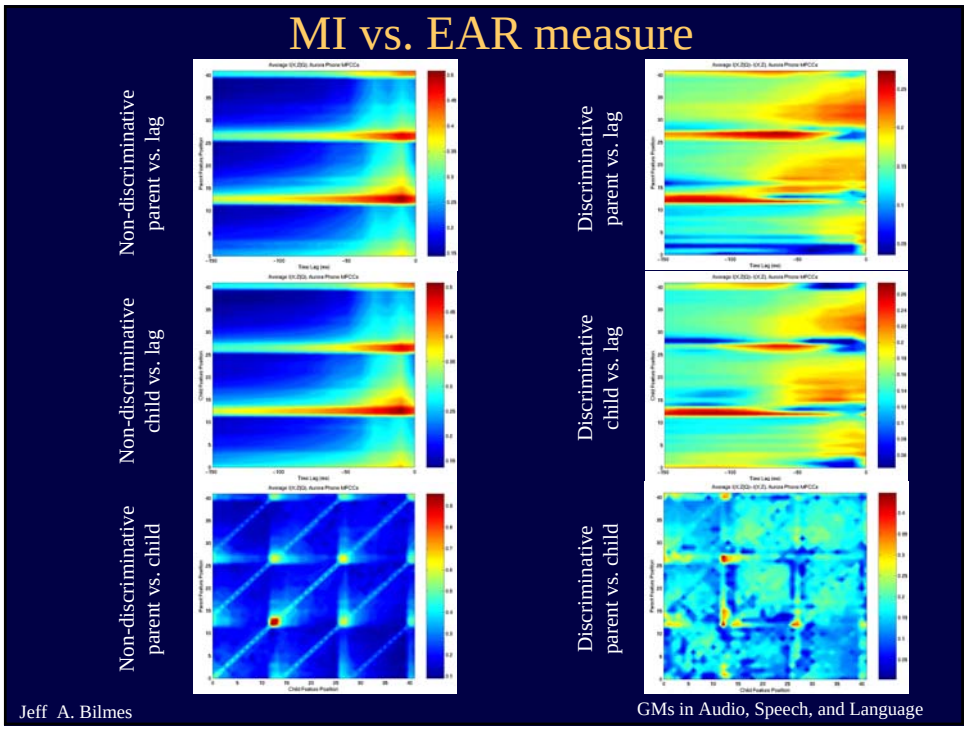
- Use Conditional Mutual Information

$$I(X;Y \mid C = c) = \sum_{x,y} p(x,y \mid c) \log \frac{p(x,y \mid c)}{p(x \mid c)\, p(y \mid c)}$$

- There are many points to compute

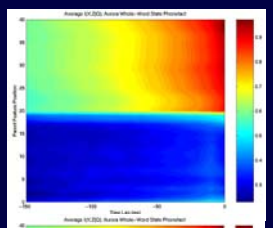



Projection Visualizations
parent-lag plot
child-lag plot
parent-child plot
i child
i parent
lag
Jeff A. Bilmes
GMs in Audio, Speech, and Language

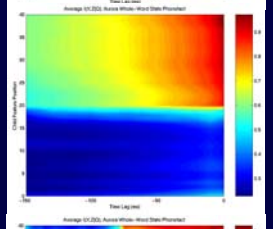

MI vs. EAR measure
Non-discriminative parent vs. lag
Non-discriminative child vs. lag
Non-discriminative parent vs. child
Discriminative parent vs. lag
Discriminative child vs. lag
Discriminative parent vs. child
Jeff A. Bilmes
GMs in Audio, Speech, and Language

MI vs. EAR measure

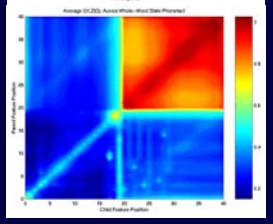
Non-Discriminative parent vs. lag



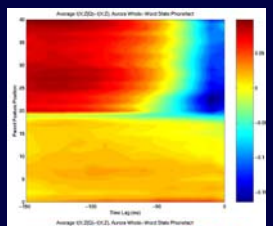
Non-Discriminative child vs. lag



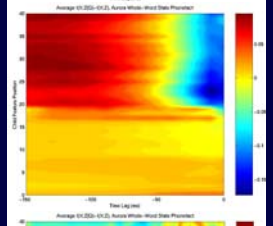
Non-Discriminative parent vs. child



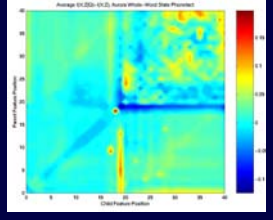
Discriminative parent vs. lag



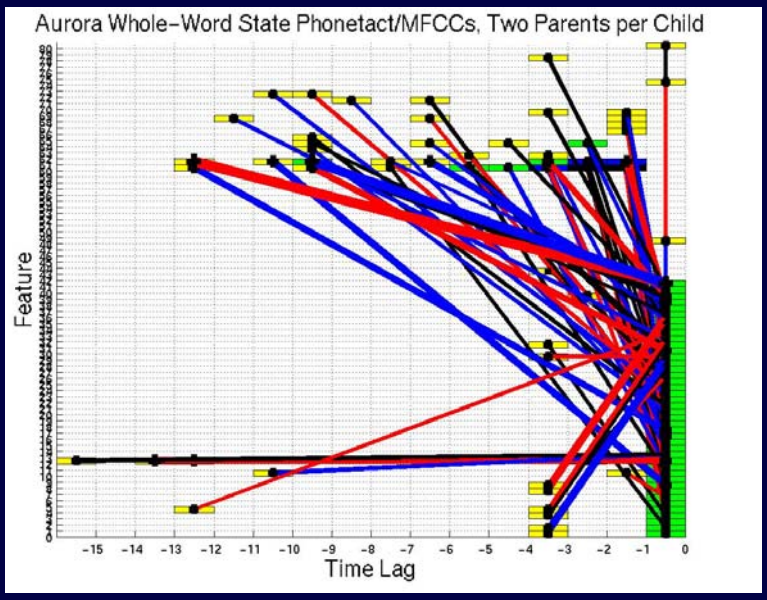
Discriminative child vs. lag



Discriminative parent vs. child



Learned



GMTK: Graphical Models Toolkit

- A GM-based software system for speech, language, and time-series modeling
- One system – Many different underlying statistical models (more than an HMM)
- Complements rather than replaces other ASR and GM systems (e.g., HTK, AT&T, ISIP, BNT, BUGS, Hugin, etc.)
- Ultimately will be open-source, freely available
- Long-term multi-year goal: improve features, computational speed, and portability.

GMTK is infrastructure

- GMTK does not solve speech and language processing problems, but provides tools to help to simplify testing modeling, and does so in novel ways.
- The space of possible solutions is huge, and its exploration has only just begun.

GMTK Features

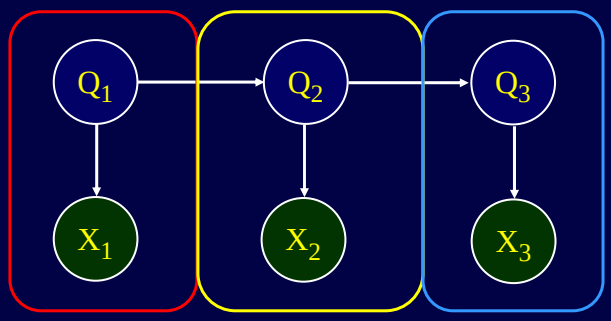
- Textual Graph Language
- Switching Parent Functionality
- Linear/Non-linear Dependencies on observations
- Arbitrary low-level parameter sharing (EM/GEM training)
- Gaussian Vanishing/Splitting algorithm.
- Decision-Tree-Based implementations of dependencies (deterministic and sparse)
- Full inference, single pass decoding possible on smaller tasks (current version)
- Sampling Methods
- Log space Exact Inference – Memory O(logT)

GMTK Structure file for HMM

```
frame : 0 {
    variable : state {
         type : discrete hidden cardinality 4000;
         switchingparents : nil;
         conditionalparents : nil using DenseCPT("pi");
    }
    variable : observation {
         type : continuous observed 0:39;
         switchingparents : nil;
         conditionalparents : state(0) using mixGaussian mapping("state2obs");
    }
}
frame : 1 {
    variable : state {
         type : discrete hidden cardinality 4000;
         switchingparents : nil;
         conditionalparents : state(-1) using DenseCPT("transitions");
    }
    variable : observation {
         type : continuous observed 0:39;
         switchingparents : nil;
         conditionalparents : state(0) using mixGaussian mapping("state2obs");
    }
}
```

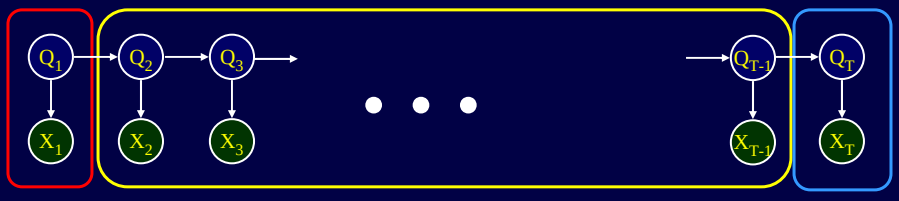
GMTK Structure file for HMM

- Structure file defines a prologue , chunk , and epilog . E.g., for the basic HMM:



GMTK Unrolled structure

- Chunk is unrolled T-size(prologue)-size(epilog) times (if 1 frame in chunk)

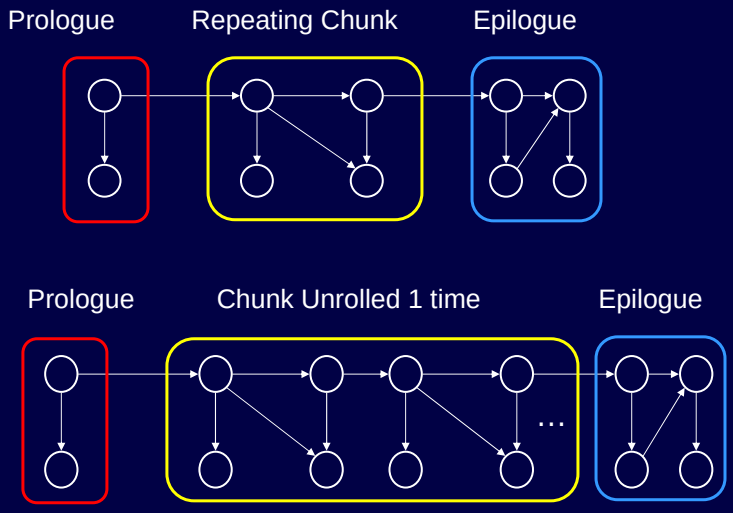


Prologue, first group of frames

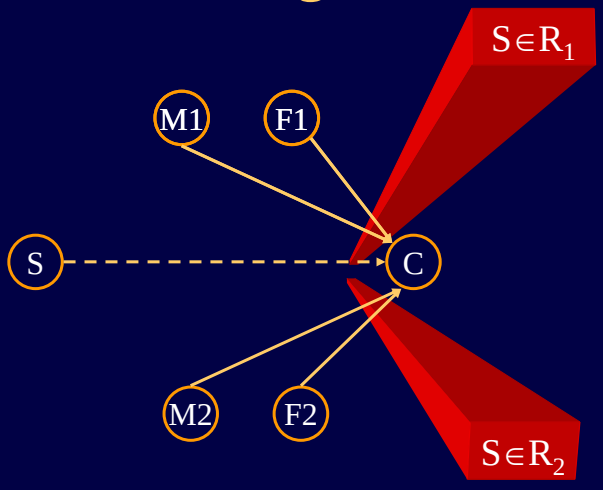
Chunk, Repeated until T frames is obtained.

Epilog, last group of frames

Multiframe Repeating Chunks



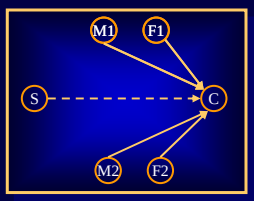
Switching Parents



$$P(C \mid M1, F1, F2, F2) = \sum_i P(C \mid M_i, F_i, S = i) P(S \in R_i)$$

GMTK Switching Structure

```
variable : S {
        type : discrete hidden cardinality 100;
        switchingparents : nil;
        conditionalparents : nil using DenseCPT("pi");
}
variable : M1 {...}
variable : F1 {...}
variable : M2 {...}
variable : F2 {...}
variable : C {
   type : discrete hidden cardinality 30;
   switchingparents : S(0) using mapping("S-mapping");
   conditionalparents :
        M1(0),F1(0) using DenseCPT("M1F1")
      | M2(0),F2(0) using DenseCPT("M2F2");
}
```



Gaussians Represented as Bayesian Networks

- Factor concentration matrix: K = U'DU
  - D = positive diagonal of conditional variances
  - U = unit upper-triangular matrix
- det(U) = 1 so det(U'DU) = det(D)
- Gaussian becomes:

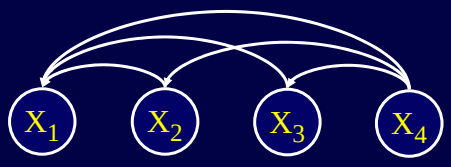
$$f(x) = \left|2\pi D\right|^{-1/2} e^{-\frac{1}{2}(x-\mu)'U'DU(x-\mu)}$$

Gaussians and Directed Models
The B matrix

$$K = U'DU = (I - B)'D(I - B)$$

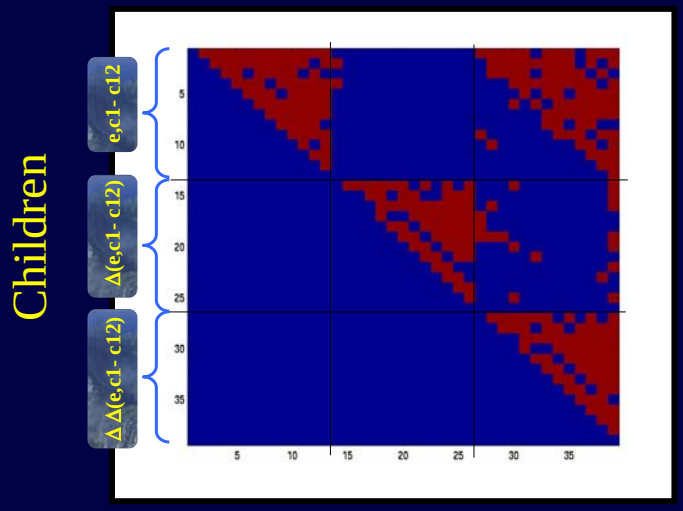
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & b_{12} & b_{13} & b_{14} \\ 0 & 0 & b_{23} & b_{24} \\ 0 & 0 & 0 & b_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{bmatrix}$$

- A Gaussian can be viewed as a directed graphical model
- FSICMs, obtained via U'DU factorization, provides the edge coefficients



$$f(x) = \prod_i f(x_i \mid x_{\pi_i})$$

Actual Sparse Covariance Matrix



GMTK Sharing & EM/GEM Training

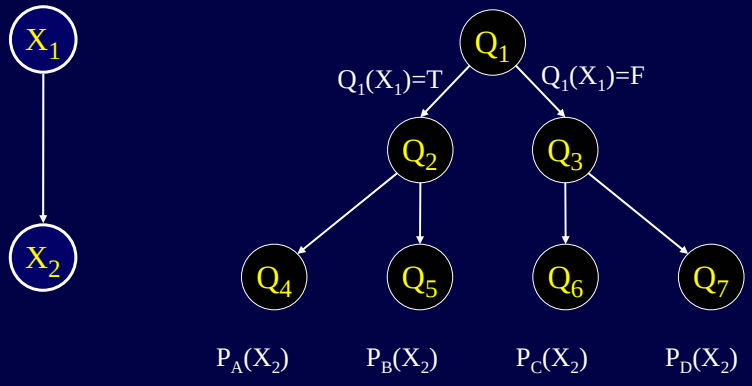
- In GMTK, Gaussians are viewed as directed graphical models.
- GMTK supports arbitrary parameter sharing:
  - Any Gaussian can share its mean, variance D, and/or its (sparse) B matrix with others.
- Normal EM training leads to a circularity
- GMTK training uses a GEM algorithm

$$(\mu^*, D^*, B^*) = \underset{\mu, D, B}{\text{argmax}}\ Q(\mu, D, B; \mu^o, D^o, B^o)$$

GMTK Splitting/Vanishing Algorithm

- Determines number Gaussian components/state
- Split Gaussian if it's component probability ("responsibility") rises above a number-of-components dependent threshold
- Vanish Gaussian if it's component probability falls below a number-of-components dependent threshold
- Use a splitting/vanishing schedule, one set of thresholds per each EM training iteration.

Decision-tree implementation of discrete dependencies



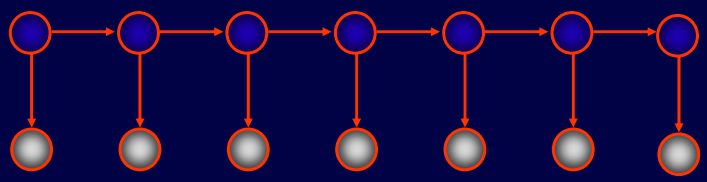
Linear and Log space exact inference

- Exact inference O(T*S) space and time complexity, S = clique state space size
- Log-space inference O(log(T)*S) space at an extra cost of a factor of log(T) time.
- Can use both linear and log space inference at same time (for optimal tradeoff).
- This is same idea as what has been called the Island Algorithm

Example: Linear-Space in HMM

$$\alpha_i(t) = \sum_j \alpha_j(t-1) a_{ji} b_i(x_t)$$

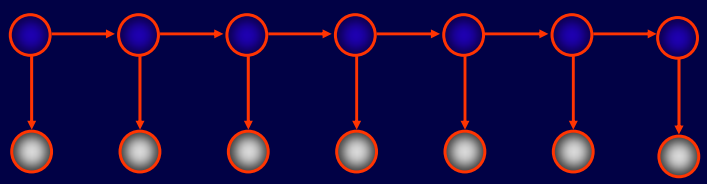
$$\beta_i(t) = \sum_j \beta_j(t+1) a_{ij} b_j(x_{t+1})$$



Example: One recursions Log Space

$$\alpha_i(t) = \sum_j \alpha_j(t-1) a_{ji} b_i(x_t)$$

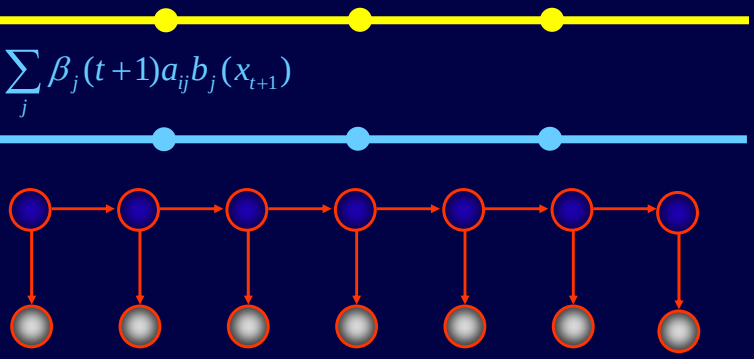
$$\beta_i(t) = \sum_j \beta_j(t+1) a_{ij} b_j(x_{t+1})$$



Example: Two recursions Log Space

$$\alpha_i(t) = \sum_j \alpha_j(t-1) a_{ji} b_i(x_t)$$

$$\beta_i(t) = \sum_j \beta_j(t+1) a_{ij} b_j(x_{t+1})$$



The GMTK Triangulation Engine (an anytime algorithm)

- User specifies an amount of time (2mins, 3 hours, 4 days, 5 weeks, etc.) to spend triangulating
- User does not worry about intricacies of graph triangulation
- Uses a "boundary algorithm" to find chunks of DBN to triangulate (UAI'2003)
- Many heuristics implemented: min-fill in, min size, min weight, maximum cardinality search, simulated annealing, exhaustive elimination, and exhaustive triangulation

Current Status

I. System available at:
   A. http://ssli.ee.washington.edu/~bilmes/gmtk
   B. ~100 pages of documentation
   C. Book chapter on use of graphical models for speech and language
   D. JHU'2001 Workshop technical report

II. GMTK Triangulation "Engine" running and ready

Exact inference in DBNs

- Triangulation in DBNs
  - Standard triangulation heuristics typically poor for DBNs since they are short and wide
  - Slice-by-slice triangulation via elimination: severely limit number of elimination orders without limiting optimal triangulation quality
  - Triangulation quality is lower-bounded by size of interface to previous (or next) slice
  - Can allow interfaces to span multiple slices, which can make interface quality much better (UAI'2003)
- Use message passing order in junction tree that respects directed deterministic dependencies (to cut down on state space)

Approximate inference in DBNs

- Standard approximate inference methods
  - Pruning as is done performed by modern speech recognition systems
  - Variational and mean-field approaches
  - Loopy belief propagation
  - Sampling, particle filtering, etc.
- All techniques for approximate inference in DBNs are relevant to the speech/language case as well.

Conclusions

- Many models and many techniques
- We have just scratched the surface, still a relatively young research area.
- Key challenges summary:
  - Explicit Control Structures
  - Structure learning
  - Fast inference techniques
  - Identifying interesting latent variables
  - Structural Discriminability

The End