Hidden Markov Models
applications and extensions

HMMs: applications and extensions

Outline

- ♢ Speech as probabilistic inference
- ♢ Speech sounds
- ♢ Word pronunciation
- ♢ Word sequences
- ♢ Biological sequence analysis
- ♢ Stochastic grammars
- ♢ Dynamic Bayesian networks

Speech as probabilistic inference

Speech signals are noisy, variable, ambiguous

What is the **most likely** word sequence, given the speech signal?
I.e., choose $Words$ to maximize $P(Words|signal)$

Use Bayes' rule:

$$P(Words|signal) = \alpha P(signal|Words) P(Words)$$

I.e., decomposes into acoustic model + language model

$Words$ are the hidden state sequence, $signal$ is the observation sequence

Phones

All human speech is composed from 40-50 phones, determined by the configuration of articulators (lips, teeth, tongue, vocal cords, air flow)

Form an intermediate level of hidden states between words and signal

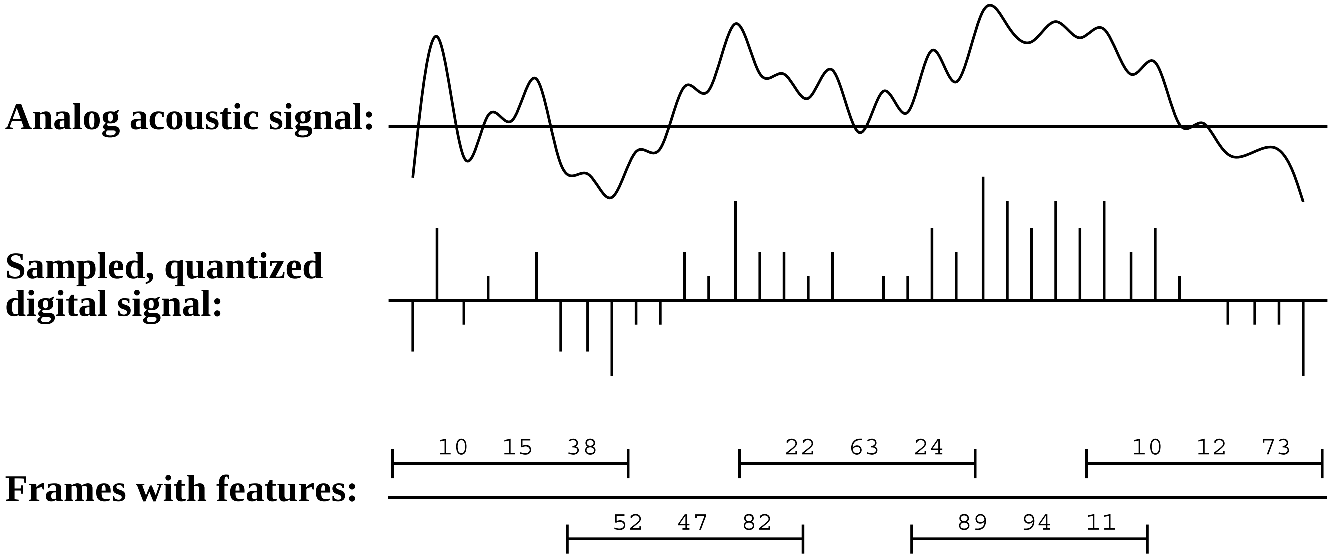
$\Rightarrow$ acoustic model $=$ pronunciation model $+$ phone model

ARPAbet designed for American English

[iy]	b**ea**t	[b]	**b**et	[p]	**p**et
[ih]	b**i**t	[ch]	**Ch**et	[r]	**r**at
[ey]	b**e**t	[d]	**d**ebt	[s]	**s**et
[ao]	b**ough**t	[hh]	**h**at	[th]	**th**ick
[ow]	b**oa**t	[hv]	**h**igh	[dh]	**th**at
[er]	B**er**t	[l]	**l**et	[w]	**w**et
[ix]	ros**e**s	[ng]	si**ng**	[en]	butt**on**
⋮	⋮	⋮	⋮	⋮	⋮

Speech sounds

Raw signal is the microphone displacement as a function of time; processed into overlapping 30ms frames, each described by features



Frame features are typically formants—peaks in the power spectrum

Phone models

Frame features in $P(features|phone)$ summarized by

– an integer in $[0 \ldots 255]$ (using vector quantization); or

– the parameters of a mixture of Gaussians

Three-state phones: each phone has three phases (Onset, Mid, End)

E.g., [t] has silent Onset, explosive Mid, hissing End

$\Rightarrow P(features|phone, phase)$

Triphone context: each phone becomes n^2 distinct phones, depending on the phones to its left and right

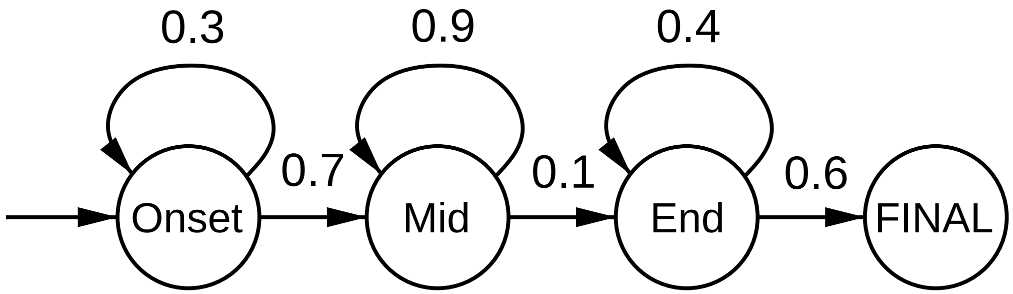
E.g., [t] in “star” is written [t(s,aa)] (different from “tar”!)

Triphones useful for handling coarticulation effects: the articulators have inertia and cannot switch instantaneously between positions

E.g., [t] in “eighth” has tongue against front teeth

Phone model example

Phone HMM for [m]:



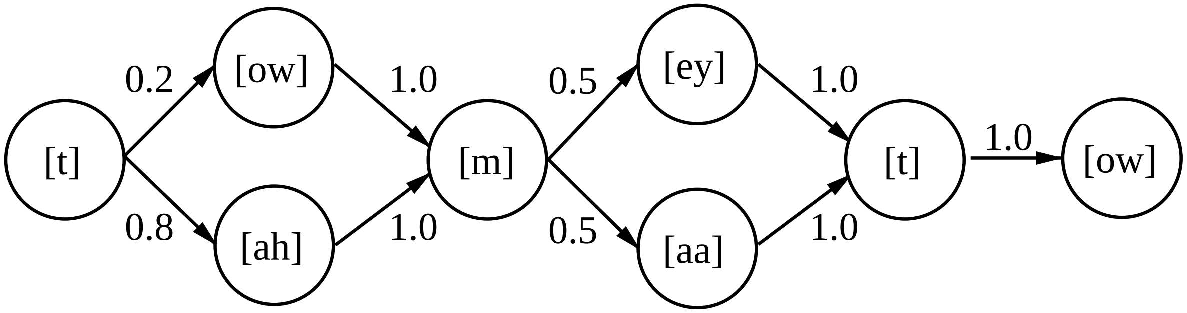
Output probabilities for the phone HMM:

Onset:	Mid:	End:
C1: 0.5	C3: 0.2	C4: 0.1
C2: 0.2	C4: 0.7	C6: 0.5
C3: 0.3	C5: 0.1	C7: 0.4

Word pronunciation models

Each word is described as a distribution over phone sequences

Distribution represented as an HMM transition model



$$P([towmeytow] | \text{“tomato”}) = P([towmaatow] | \text{“tomato”}) = 0.1$$
$$P([tahmeytow] | \text{“tomato”}) = P([tahmaatow] | \text{“tomato”}) = 0.4$$

Structure is created manually, transition probabilities learned from data

Isolated words

Phone models + word models fix likelihood $P(e_{1:t}|word)$ for isolated word

$$P(word|e_{1:t}) = \alpha P(e_{1:t}|word)P(word)$$

Prior probability $P(word)$ obtained by counting word frequencies

$P(e_{1:t}|word)$ can be computed recursively: define

$$\boldsymbol{\ell}_{1:t} = \mathbf{P}(\mathbf{X}_t, \mathbf{e}_{1:t})$$

and use the recursive update

$$\boldsymbol{\ell}_{1:t+1} = \textsc{Forward}(\ell_{1:t}, \mathbf{e}_{t+1})$$

and then $P(e_{1:t}|word) = \sum_{\mathbf{x}_t} \boldsymbol{\ell}_{1:t}(\mathbf{x}_t)$

Isolated-word dictation systems with training reach 95–99% accuracy

Continuous speech

Not just a sequence of isolated-word recognition problems!

- Adjacent words highly correlated
- Sequence of most likely words $\neq$ most likely sequence of words
- Segmentation: there are few gaps in speech
- Cross-word coarticulation—e.g., “next thing”

Continuous speech systems manage 60–80% accuracy on a good day

Language model

Prior probability of a word sequence is given by chain rule:

$$P(w_1 \cdots w_n) = \prod_{i=1}^{n} P(w_i | w_1 \cdots w_{i-1})$$

Bigram model:

$$P(w_i | w_1 \cdots w_{i-1}) \approx P(w_i | w_{i-1})$$

Train by counting all word pairs in a large text corpus

More sophisticated models (trigrams, grammars, etc.) help a little bit

Combined HMM

States of the combined language+word+phone model are labelled by the word we're in + the phone in that word + the phone state in that phone

Viterbi algorithm finds the most likely **phone state** sequence

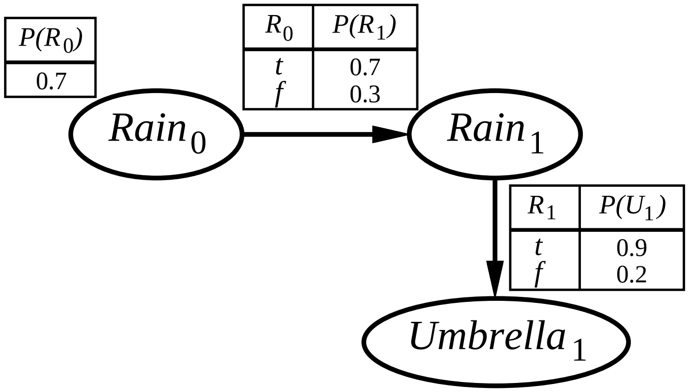
Does segmentation by considering all possible word sequences and boundaries

Doesn't always give the most likely word sequence because each word sequence is the sum over many state sequences

Jelinek invented A* in 1969 a way to find most likely word sequence
where "step cost" is $-\log P(w_i|w_{i-1})$

Dynamic Bayesian networks

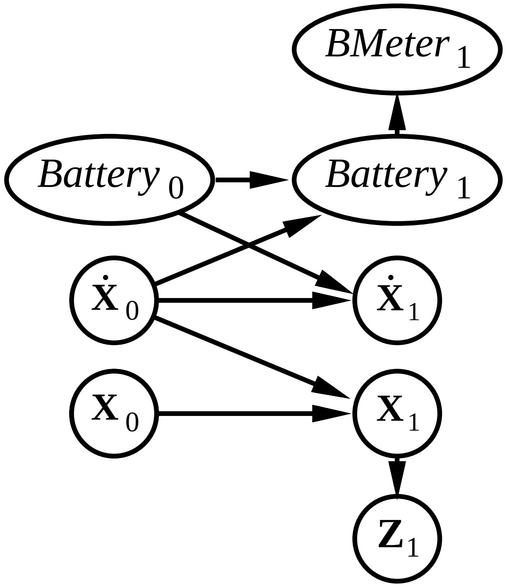
$\mathbf{X}_t$, $\mathbf{E}_t$ contain arbitrarily many variables in a replicated Bayes net



$P(R_0)$
0.7

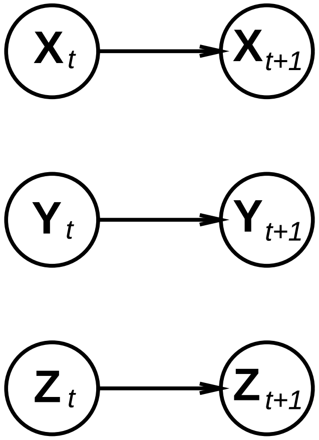
R_0	$P(R_1)$
t	0.7
f	0.3

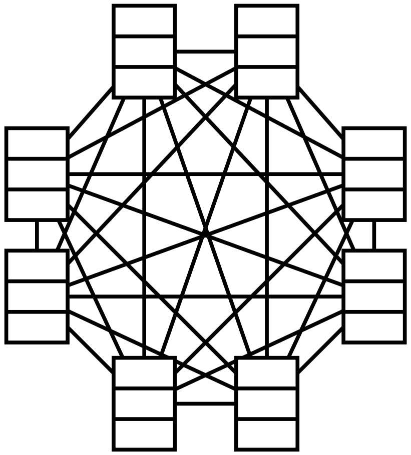
R_1	$P(U_1)$
t	0.9
f	0.2



DBNs vs. HMMs

Every HMM is a single-variable DBN; every discrete DBN is an HMM



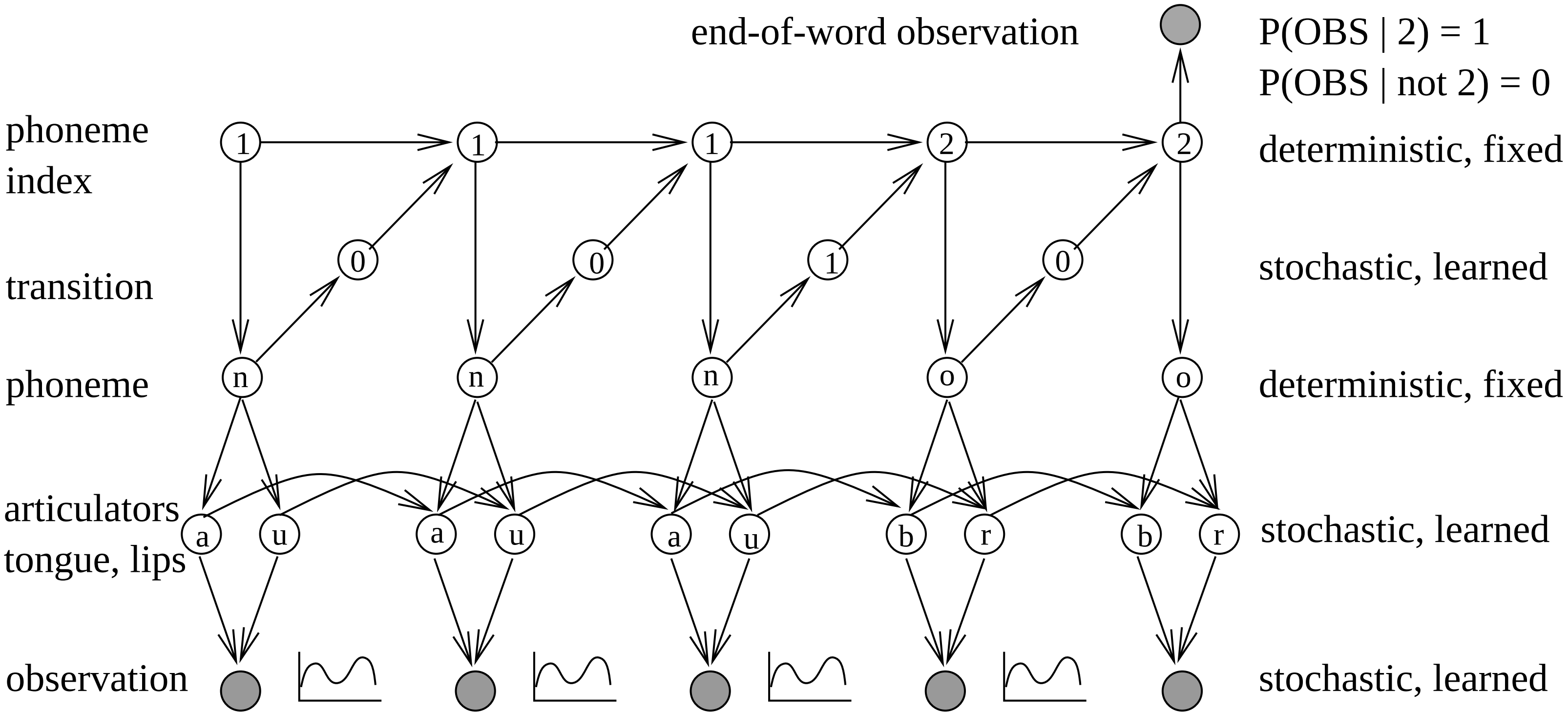


Sparse dependencies $\Rightarrow$ exponentially fewer parameters;

e.g., 20 state variables, three parents each

DBN has $20 \times 2^3 = 160$ parameters, HMM has $2^{20} \times 2^{20} \approx 10^{12}$

DBNs for speech recognition



Also easy to add variables for, e.g., gender, accent, speed.

Zweig and Russell (1998) show up to 40% error reduction over HMMs

The "profile" HMMs (pHMMs)

Define a structure (allowed transitions) over states with cardinality n. Note, $\mathcal{O}(n^2)$ parameters can be reduced to linear. . .)

Substitutions: match states (boxes). Note, level 1 implements already a position specific scoring.

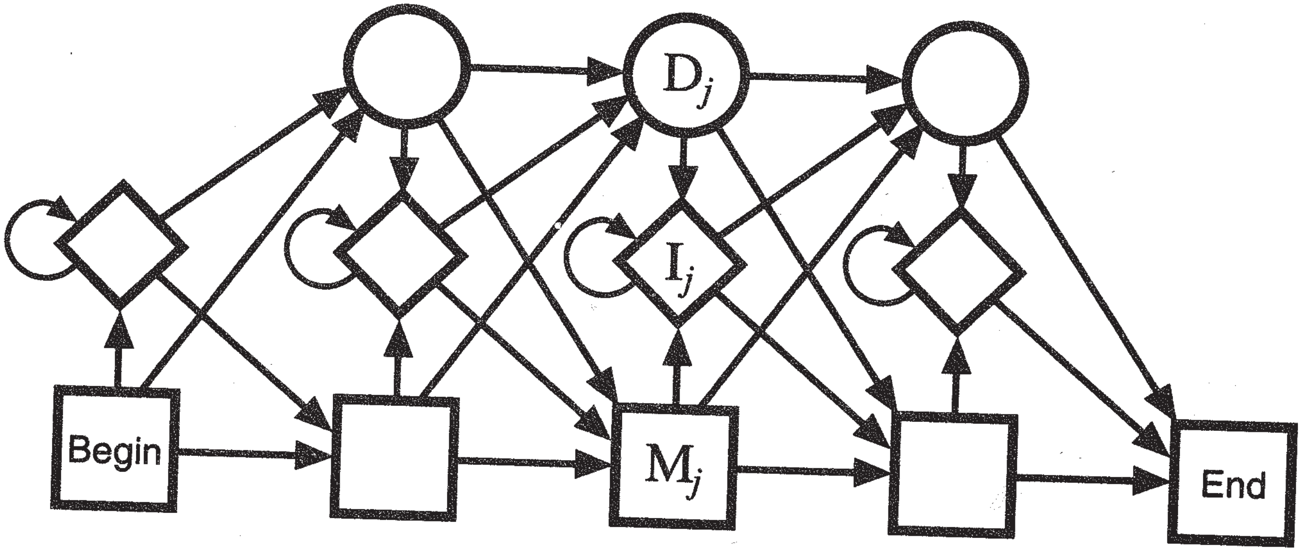
Inserts: insert states (diamonds). Note that length distribution of inserts follows a geometric distribution with parameter p of probability of stay (mean $p/(1-p)$ and variance $p/(1-p)^2$).

Deletes: transitions "jumping" over match states. Problem: high number of parameters. Solution: further parametric restriction over transition probabilities using silent delete states (circles). Note the possible reduction of $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ representing a position specific gap length penalty or even to 1 representing a gap length penalty.

Note that delete states are so called silent/null states without emission. If there are no loops as in pHMMs → emulate their effect in Viterbi/forward/backward algs treating separately the probability of transitions without emissions, e.g. accumulating upward $f_l(i+1)+=\sum_k f_k(i+1)a_{kl}$ through silent states $k<l$.

The ”profile” HMMs (pHMMs) II.

The profile HMM.



Usage: 1, exploration/visualization of a sequence family 2, deciding membership (for transferring annotations about functionality/structure) 3, (the most probable) multiple alignment

Application: see Pfam.

Gene finding:GENSCAN

Semihidden HMM a state can emit words with arbitrary length distribution L_S and symbols $Y_{S,l}$ (not just a symbol or words with length following a geometric distribution). A parse ϕ is a sequence of states and corresponding lengths (partition of observation is not trivial in such case!). HMM algorithms are more complex.

Application: parse of a DNA-segment with Viterbi.

Burge(1997)/EG:GENSCAN, human

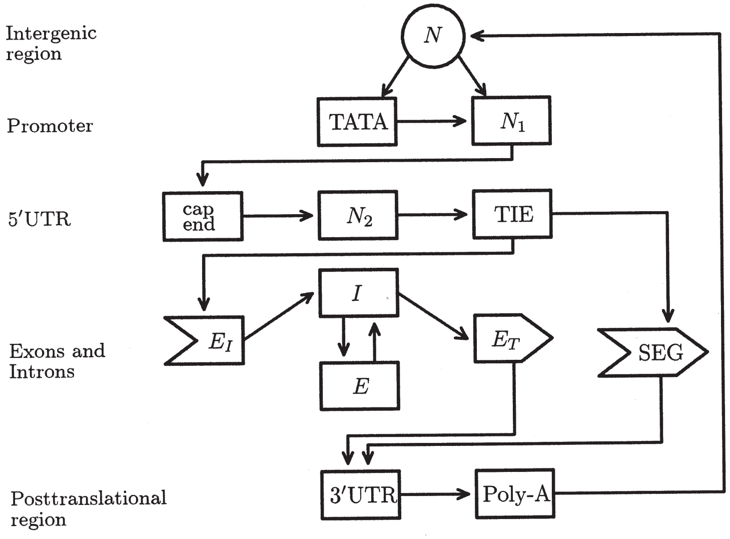
Recall: what is a gene? (here we follow a protein-coding interpretation)

The training data: 380 genes, 142 single-exon genes, 1492 exons, coding region of 1619 genes

Gene finding:GENSCAN

Structural elements(of a protein coding gene): see slides.

1. Upstream region: promoter region: **TATA box**: present in 70% of genes at 28-34 bases upstream from the start of transcription.
2. **5' untranslated region** (5'UTR): follows the promoter starting with the **cap end** region (8 bases) and ending with **translation initiation end** (TIE) (18 bases).
3. $Exon - [intron - exon]^*$: recall intron types and structure
4. **3' untranslated region** (3'UTR) contains one or more **Poly-A** signal (6 bases).



Gene finding:GENSCAN II.

The **transition** probabilities are estimated from the data (to TATA, to SEG and to multi-exon).

Intergenic region: Distance between genes is modeled by a geometric distribution with mean $p/1-p = |genome|/|genes|$ and the sequence is generated with a fifth-order MC with parameters $3 \cdot 4^5$ called **intergenic null model** (INM).

TATA box is modeled with a 15-base weight matrix (independent multinomials). N_1 is from the INM with length distributed uniformly from 28 to 34. **Cap end** is modeled with an 8-base weight matrix. N_1 is from the INM with length from a geometric distribution with mean 735 bases. **TIE** is modeled with a 18-base weight matrix.

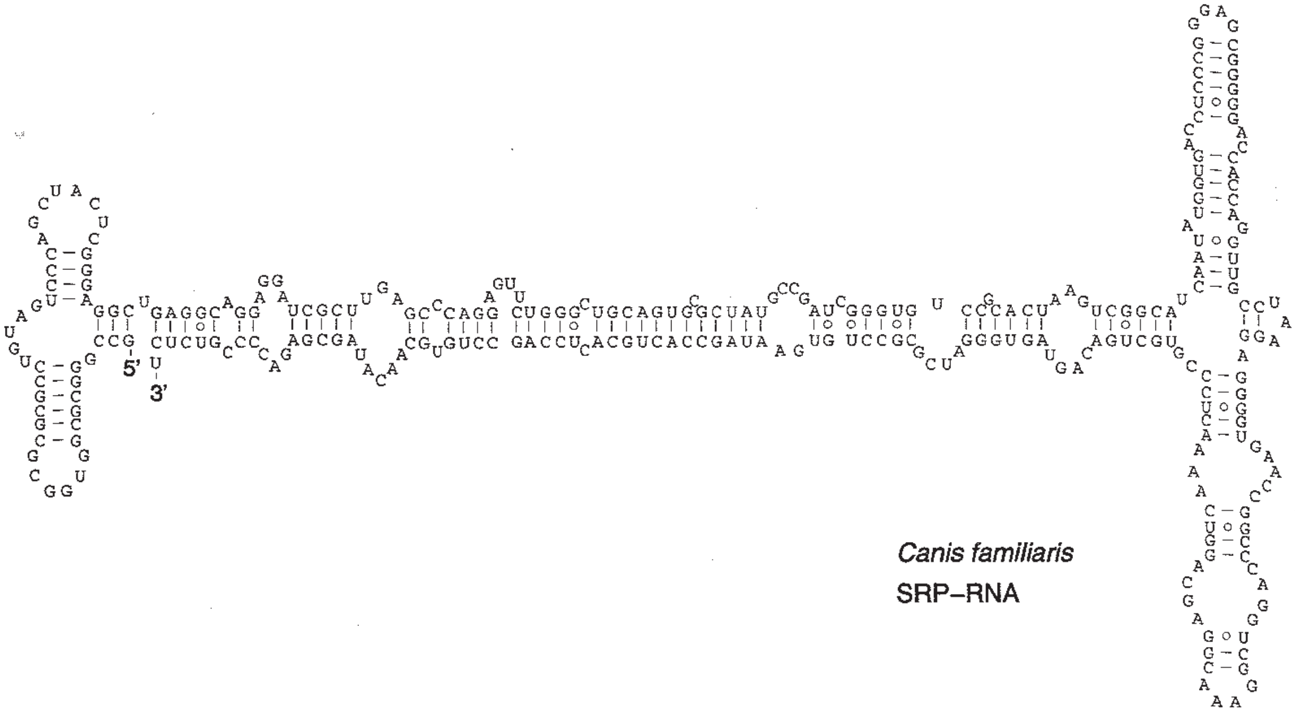
Single exon gene (SEG) is modeled with a nonhomogeneous (3-phase) fifth-order MC generating first the start codon atg and ending with the three stop codons taa, tag, tga. Length follows the empirical distribution.

Multiexon gene is modeled with the SEG model for the exons. The length of the introns are modeled with empirical distribution independently for initial, internal and terminal introns. The intron sequence generation starts with splitting a random codon with 1/3 probability to 0/3, 1/2 or 2/1. This prefix starts the intron, then the donor splice signal is modeled with a decomposed weight matrix with length 6, then the INM generates the intron, finally the acceptor splice signal is again modeled with a decomposed weight matrix with length 20, which is closed with postfix part of the splitted codon.

The **3'UTR** is modeled with the INM with geometric length of mean 450. The **Poly-A** is modeled with a 6-base weight matrix.

RNA I

RNA is single stranded sequence of bases A, U, C, G, but base pairs arise such as A-U, G-C (canonical) or non-canonical pairs such as G-U, which are relatively stable as well. ⇒ An RNA strand has complex structure because of (linearly) distant, but paired bases. RNA is not just a "messenger", but effector (autocatalytic RNAs) (⇒ "RNA world" hypothesis).



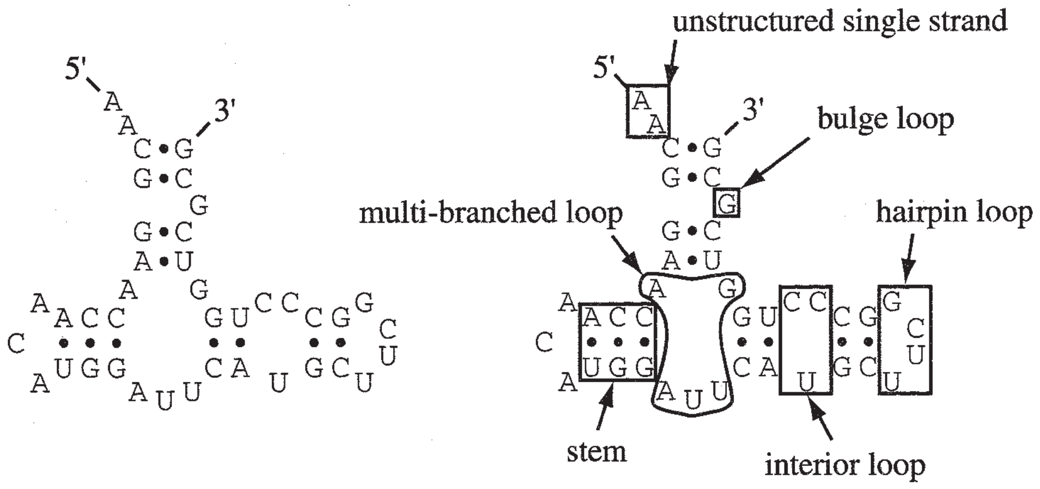
Canis familiaris
SRP–RNA

RNA II

Consecutive stacked base pairs called *stem* form A-form double helix (distorted by non-canonical pairs). A stem is surrounded by single stranded subsequences called *loops* (bulge/interior/hairpin and multibranch loops). These form the secondary structure of the RNA sequence.

Type of interactions:

1. **nested**: $(i,j),(i',j')$ pairs are nested-pairs if not related (e.g. $i < j < i' < j'$) or nested (e.g. $i < i' < j' < j$),
2. **non-nested**: base-pairs: copies, meta/reversed-copies (1%).



Grammars

Goal: definition of a given set of words (language $\mathcal{L}$) over a finite alphabet Σ.

Generative/transformational grammars: Members of the language can be derived using *rewrite rules* containing *terminal* and *nonterminal* symbols (denoted with small and capital letters). *Parsing* consists of the reconstruction of a derivation/parse tree (alignment).

Questions:

1. parsing: find parse T resulting in terminal sequence x

2. membership: $x \in \mathcal{L}^G$ or is there any parse T resulting in terminal sequence x

Chomsky hierarchy of grammars

Grammar	Rule	Automaton	Parsing	Language
regular*	$W \to aW$	FSA	linear	a reg.expression
context-free	$W \to \beta$	push-down	polynomial	palindromes
context-sensitive**	$\alpha_1 W \alpha_2 \to \alpha_1 \beta \alpha_2$	linear bounded	exponential	copies
unrestricted		Turing machine (TM)	semidecidable	$KB - FOL \models \alpha$
-	-			halting TMs

(*:right/left, with/without ϵ;**:nondecreasing)

Stochastic grammars

Rewrite rules in grammar G have application probabilities (θ denotes their vector).

Questions (T_x denotes parse tree with terminal sequence x):

1. parsing: $T_x^* = \arg\max_{T_x} p(T_x|\theta, G)$
2. membership: $p(x|\theta, G) = \sum_{T_x} p(T_x|\theta, G)$
3. parameter learning: $\theta^* = \arg\max_{\theta} p(x^{(1)}, \ldots, x^{(n)}|\theta, G)$
4. posterior decoding:
   $p(W \rightarrow x_{i:j}|x, \theta, G) = \sum_{T_x} p(T_x|\theta, G)$ 1($x_{i:j}$ is generated from W in parse tree T_x")

SCFG algorithms

Assume: M nonterminals ($W = W_1, \ldots, W_M$), Chomsky normal form ($W_v \rightarrow W_y W_z$ or ($W_v \rightarrow a$) with transition and emission probabilities $t_v(y,z)$ and $e_v(a)$

The **inside** algorithm computes the probability of sequence x $p(x)$ summing over all possible derivation (parse tree).

Idea: calculate recursively the probability $\alpha(i,j,v)$ of a parse subtree rooted at nonterminal W_v for subsequence $x_{i:j}$ for all i,j,v.

Algorithm 1 Algorithm: inside

```
Require: SCFG,x
Ensure: $p(x|SCFG)$
  Ini: i=1 to L, v=1 to M: $\alpha(i,i,v) = e_v(x_i)$
  for i=1 to L-1 do {length}
    for j=1 to L-i do {starting positions}
      for v=1 to M do {states}
        $\alpha(j,j+i,v) = \sum_{y=1}^{M} \sum_{z=1}^{M} \sum_{k=j}^{j+i} \alpha(j,k,y)\alpha(k+1,j+i,z)t_v(y,z)$
  End: $p(x|SCFG) = \alpha(1,L,1)$
```

The **outside** algorithm computes a probability called $\beta(i,j,v)$ of a complete parse tree for sequence x, excluding subtrees with W_v nonterminal and $x_{i:j}$ leaves.

The optimal parse tree can be found by the **Cocke-Younger-Kasami (CYK)** algorithm: same as inside with $\max_{y,z,k}$ instead of $\sum_{y,z,k}$ and with pointers for backtracking.

HMMs/SFSAs/SRGs versus SCFGs

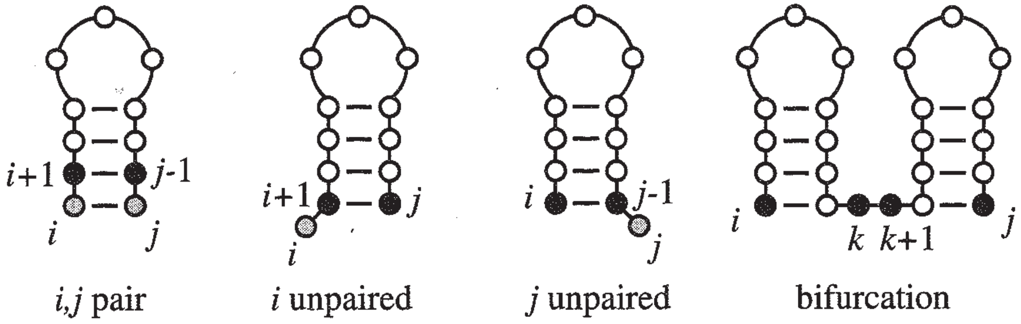
The same questions for stochastic context free grammars (SCFGs) modeling RNA:
(X^h/X^o hidden/observed variables)

Goal	stochastic regular grammars	stochastic context-free grammars
Explanation:$p(X^h \| X^o, \theta^M, M)$	alignment: Viterbi	parse tree: CYK
Matching:$p(X^o \| \theta^M, M)$	p(sequence): forward alg.	p(seq.): inside alg.
Canonical model class:$M \in \mathcal{M}$	profile HMMs (length)	covariance models
Imputation-based parameter learning:θ^M	Viterbi-based	CYK-based
EM-based parameter learning:θ^M	forward-backward	inside-outside
Time complexity	$\mathcal{O}(LM^2)$	$\mathcal{O}(L^3M^3)$
Space complexity	$\mathcal{O}(LM)$	$\mathcal{O}(L^2M)$

Note that SCFG models allows a more powerful representation of a distribution of homologous sequences than HMMs (e.g. allowing palindrome constraints) or phylogenetic tree with i.i. substitution stochastic process assumption.

PCFG:Covariance model I

An SCFG model of RNA folding based on four types of recursive extension (paired, left-unpaired, right-unpaired, bifurcation) (Nussinov)



$$S \rightarrow aSu|cSg|gSc|uSa \text{ (paired)} \quad (1)$$

$$S \rightarrow aS|cS|gS|uS \text{ (left − unpaired)} \quad (2)$$

$$S \rightarrow Su|Sg|Sc|Sa \text{ (right − unpaired)} \quad (3)$$

$$S \rightarrow SS \text{ (bifurcation)} \quad (4)$$

PCFG:Covariance model I

A generic stem model with six states (W denotes any states):

$$\begin{aligned}
P &\rightarrow aWa|\ldots \;\; (\text{pairwise}, 16) && (5)\\
L &\rightarrow aW|\ldots \;\; (\text{leftwise}, 4) && (6)\\
R &\rightarrow Wa|\ldots \;\; (\text{rightwise}, 4) && (7)\\
B &\rightarrow SS \;\; (\text{bifurcation}) && (8)\\
S &\rightarrow W \;\; (\text{start}) && (9)\\
E &\rightarrow \epsilon \;\; (\text{end}) && (10)
\end{aligned}$$

Summary

Since the mid-1970s, speech recognition has been formulated as probabilistic inference

Evidence = speech signal, hidden variables = word and phone sequences

"Context" effects (coarticulation etc.) are handled by augmenting state

Variability in human speech (speed, timbre, etc., etc.) and background noise make continuous speech recognition in real settings an open problem

The same technology could be applied and now mainly used in biomedical sequence analysis.