Tutorial Philosophy

- Questions?
- Ideas?
- Confusion?
- Suggestions?
- Please ask.
Detailed Outline

1. Properties
   a) Overview and Motivation
   b) GM Types and Constructs
   c) Theory and Practice of Dynamic GMs
   d) Explicit Temporal Structures

2. Specific Models
   a) GMs in Speech
   b) GMs in Language
   c) GMs in Machine Translation

3. Toolkits

4. Discussion/Open Questions
Cause and Effect

• Fact: At least 196 cows died in Thailand, Jan/Feb 2004, as have 16 people.
• Consequence: Canadian officials in April 2004 killed 19 million birds in British Columbia (chickens, ducks, geese, etc.)
• Possible cause I: Original deaths due to avian influenza (H5N1 or bird flu)
• Possible cause II: They died of old age!
Cause and Effect

- Simple directed graphs can be used.
- Directed edges go from **parent** (possible cause) to **child** (possible effect)

![Diagram showing cause and effect relationships: Bird Flu and Old Age as parents to Deaths, which in turn is a parent to Canadian Action.](image-url)
Cause and Effect

• Quantities of interest:

Computing Probabilities:

Examples:

Pr( Deaths | Flu and Old )
Pr( Deaths | Old )
Pr( Bird Flu | Canadian Action )

Asking Questions:

Examples:

• In general, does old age increase the chance that a cow has contracted bird flu (if at all)?
• If we know the action by Canada occurred, does having bird flu decrease the chance that it was old when it died?

Very simple scenario, with obvious answers to questions. What happens with more complicated real-world problems?
Realistic Domains

- Graph represents relational structure of domain.
- Regardless of graph complexity, same set of algorithms used to compute desirable quantities and answer all questions.

- Example: Is $\Pr(A|B)$ larger or smaller than $\Pr(A|B,C)$?
- Graphs are natural way to explain complex situations on an intuitive visual level.
- Graphs are used everywhere, for many different purposes. Here they have a specific meaning: Random variables and conditional independence.
Graphical Models (GMs)

- 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, factorization)

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 families of probabilistic models and their properties.
Graphical Models (GMs)

GMs give us (cont):

IV. **Approximation**: Methods to explore systems of approximation and their implications. E.g., what are the consequences of a (perhaps known to be) wrong assumption?

V. **Data-base**: Provide a probabilistic “database” and corresponding “search algorithms” for making queries about properties in such model families.
A Pause For Questions

- Questions?
- Ideas?
- Confusion?
- Suggestions?
- Please ask.
Outline

1. Properties
   a) Overview and Motivation
   b) GM Types and Constructs
   c) Theory and Practice of Dynamic GMs
   d) Explicit Temporal Structures

2. Specific Models
   a) GMs in Speech
   b) GMs in Language
   c) GMs in Machine Translation

3. Toolkits

4. Discussion/Open Questions
Three Important Graph Types

- Three (of many) types of GM
  - Bayesian Networks
  - Undirected Graphical Models
  - Factor Graphs
- Each conveys a different family
- Each type of graph conveys factorization of probability distributions in some way.
Factorization and Conditional Independence

- Factorization of probability distributions usually (but not always) implies some form of conditional independence.

- Conditional Independence:

  \[ X_A \perp X_B \mid X_C \]

- Holds if and only if there are functions \( f() \) and \( g() \) such that:

  \[ p(x_A, x_B, x_C) = g(x_A, x_C)h(x_B, x_C) \]

Notation: \( X_A \) is a set of random variables indexed by the set \( A \). So if \( A = \{1,2,4\} \), then \( X_A = \{X_1, X_2, X_4\} \).
Bayesian Networks

- Only one type of graphical model (many others exist)
- Compact representation: factors of probabilities of children given parents.

Sub-family specification:
- Directed acyclic graph (DAG)
- Nodes - random variables
- Edges - direct “influence”

Together:
- Defines a unique distribution in a factored form

\[ P(B, E, A, C, R) = P(B)P(E)P(A | B, E)P(R | E)P(C | A) \]
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 that is **blocked**. Node $v$ is blocked if either:
    1. $\rightarrow v \rightarrow$ 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”
  - A variable is conditionally independent of its non-descendants given its parents
- See Lauritzen ’96 for other properties.
- Distribution factors as product of conditional probability distributions of child given its parents $P(c|\text{parents})$
Three Basic Cases

- Graphs encode conditional independence via factorization.
A probabilistic data-base

• What are the implicit assumptions made by a given statistical model? Can be complicated.

\[ p(X, Y, Z, A, B, C, D, E) = p(Y \mid D) p(D \mid C, E) p(E \mid Z) p(C \mid B) p(B \mid A) p(X \mid A) p(A) p(Z) \]

Is \( X \perp\!\!\!\!\!\!\!\!\!\!\perp Z \mid Y \)??
A probabilistic data-base

- GMs can help illuminate the answer.

\[ p(X,Y,Z,A,B,C,D,E) = p(Y \mid D)p(D \mid C,E)p(E \mid Z)p(C \mid B)p(B \mid A)p(X \mid A)p(A)p(Z) \]

\[ X \perp Z \mid Y \quad ??? \]

\[ \text{No!!} \]
Hidden Variables

- Hidden variables can introduce significant capabilities for probabilistic queries (called inference).

\[
p(v_3 \mid v_1) = \sum_{v_2} p(v_3 \mid v_2) p(v_2 \mid v_1)
\]

\[
p(v_3 \mid v_1, v_2) = p(v_3 \mid v_2)
\]
Nodes as **Inputs** and/or **Outputs**

- Variables in a GM can be both an input or an output variable at different times (unlike a neural network). This can be very useful.

\[
p(v_3 | v_1) \quad p(v_1 | v_3)
\]
Undirected GMs (UGMs)

• 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)
UGMs (Markov Random Fields)

• Example independence:

\[ \{V_5, V_6\} \perp \{V_2, V_3\} | V_4 \]

• Factorization

\[
P(V_1, V_2, V_3, V_4, V_5, V_6) = f(V_1, V_4) f(V_4, V_5) f(V_5, V_6) f(V_2, V_4, V_3)
\]
Example: Undirected GMs (UGMs)

\[ P(X) = \frac{1}{Z} \prod_{c} \Psi_{X_c}(X_c) \]

c = **cliques** (completely connected nodes) in graph.

If Gibbs:
\[ = \frac{1}{Z} \exp\left\{ - \frac{1}{k_0 T} U(x) \right\} \]
\[ U(x) = \sum_{c} H_{X_c}(x_c) \]

Semantics: Simple Separation
\[ X_A \cancel{\parallel} X_B | X_C \] if \( X_C \) separates \( X_A \) from \( X_B \) in graph.

Clique are: \{ (W,X), (X,Z), (Z,Y), (Y,W) \}

UGMs are the same as MRFs
Directed and Undirected models represent different families

- The classic examples:
Factor Graphs

• Graph represents all possible factorizations of a distribution.
• Simplest example:

\[ P(V_1, V_2, V_3) = f(V_1, V_2)f(V_2, V_3)f(V_3, V_1) \]
A Pause For Questions

• Questions?
• Ideas?
• Confusion?
• Suggestions?
• Please ask.
Outline

1. Properties
   a) Overview and Motivation
   b) GM Types and Constructs
   c) Theory and Practice of Dynamic GMs
   d) Explicit Temporal Structures

2. Specific Models
   a) GMs in Speech
   b) GMs in Language
   c) GMs in Machine Translation

3. Toolkits

4. Discussion/Open Questions
Dynamic Bayesian Networks (DBNs)

- Most appropriate for speech/language.
- DBNs are Bayesian Networks over time.
- Specified using a “rolled up” template.
- In unrolled DBN, all variables sharing same origin in template have tied parameters.
- Allows for specifying graph over arbitrary length series.
Hidden Markov Models

- HMMs are DBNs.

- Corresponds to template:

Note: These graphs is **not** stochastic finite state automata, as in:
Dynamic Bayesian Networks

- More generally, DBN specifies template to be unrolled:

- Intra-slice edges
- Inter-slice edges

Unrolled DBN, 3 Times

DBN Template
General Probabilistic Inference

\[ P(x_1 \mid x_6) = \frac{p(x_1, x_6)}{p(x_6)} \]

\[ P(x_1, x_6) = \sum_{x_{2:5}} p(x_{1:6}) \]

Exploit local structure to provide for efficient inference \( O(r^3) \) rather than \( O(r^6) \)
(variable elimination algorithm)

\[ \sum_{x_{2:5}} P(x_1, x_2, x_3, x_4, x_5, x_6) \]

\[ = \sum_{x_{25}} P(x_1) P(x_2 \mid x_1) P(x_3 \mid x_1) P(x_4 \mid x_2) P(x_6 \mid x_2, x_5) P(x_5 \mid x_3) \]

\[ = \sum_{x_2} P(x_1) P(x_2 \mid x_1) \sum_{x_3} P(x_3 \mid x_1) \sum_{x_5} P(x_6 \mid x_2, x_5) P(x_5 \mid x_3) \sum_{x_4} P(x_4 \mid x_2) \]

- Way in which sums are distributed into products corresponds to different ways of running the junction tree algorithm (generalization of Baum Welsh)
Moralization & Triangulation, lead to Junction Tree

Original

Moralized

Complexity of inference:

\[ O(\sum_c s(c)) \]

\[ s(c) \approx r^{|c|} \]

Junction Tree
Moralization & Triangulation

• Moralization
  – why ok? More edges, fewer independence assumptions, bigger family
  – why needed? So UGM doesn’t violate BN semantics, summations include parents in cliques.

• Triangulation
  – why ok? Fewer independencies -> bigger family
  – why needed? To get a decomposable graph, where junction tree (with running intersection property) exists and message passing algorithm is correct (i.e., local consistency implies global consistency)
Inference: Message Passing in JT

Now, all cliques equal the joint of their variables and any evidence (observations), e.g., $P(A,B,C,D,\text{Obs})$, $P(B,C,D,F,\text{Obs})$, etc.

1. Collect Evidence Phase
2. Distribute Evidence Phase
Inference in Dynamic Models

- Inference procedures are equivalent to forming a “dynamic” Junction Tree
- Generalizes Forward/Backward (Baum/Welch) procedure in HMMs
- Dynamic Junction Tree is much wider than higher.
Inference in Dynamic Models

Unrolled DBN, 3 Times

DBN Template

Intra-slice edges

Inter-slice edges
Inference in Dynamic Models

- Intra-slice edges
- Inter-slice edges
- Moralization edges

Moralized DBN

Graphical Models
Inference in Dynamic Models

- Boundary between slices defines graph partitions to triangulate

**Intra-slice edges**

**Inter-slice edges**

**Moralization edges**

**Compulsory Interface edges**

**Triangulation Edges**

DBN Partition

Resulting Triangulated DBN Partition
Inference in Dynamic Models

• Each partition is stitched together to create what is guaranteed to be a triangulated DBN

Resulting Triangulated DBN

- Intra-slice edges
- Inter-slice edges
- Moralization edges
- Compulsory Interface edges
- Triangulation Edges
Inference in Dynamic Models

• Resulting Junction Tree (note: better ones exist, see Bilmes&Bartels UAI’03 on triangulating DBNs)

• Junction “Tree” in this case is a Markov chain
• Not dissimilar to Hidden Markov Model alpha-recursion, but here the “cliques” oscillate between two forms.
Inference is Hard

- NP complete (exponentially difficult) to perform inference.
- Goal: find small cliques, since complexity is exponential in clique size (also hard, NP)
- Approximate inference schemes exist for harder problems
  - variational approaches
  - sampling techniques (MCMC, Gibbs, etc.)
  - loopy belief propagation (LDPC & turbo codes)
  - Pruning procedures
“Learning” Graphical Models

• Five scenarios for learning:
  1. Structure known, no hidden variables
  2. Structure known, hidden variables
  3. Structure unknown, no hidden variables
  4. Structure unknown, edges unknown over known hidden variables
  5. Structure unknown, unknown set of hidden variables.

• Typically, we need to do 5 for speech/language/machine translation recognition.
A Pause For Questions

• Questions?
• Ideas?
• Confusion?
• Suggestions?
• Please ask.
Outline

1. Properties
   a) Overview and Motivation
   b) GM Types and Constructs
   c) Theory and Practice of Dynamic GMs
   d) Explicit Temporal Structures

2. Specific Models
   a) GMs in Speech
   b) GMs in Language
   c) GMs in Machine Translation

3. Toolkits

4. Discussion/Open Questions
Why Graphical Models for Speech and Language Processing

- Expressive concise way to describe properties of families of distributions
- **Rapid** movement from novel idea to implementation (with right toolkit) – All graphs utilize *exactly same inference algorithm*! Researcher concentrates on model and can stay focused on domain.
- GMs include many standard techniques but GM space is hardly explored.
- Dynamic graphical models can represent important structure in “natural” time signals but ignore what is unimportant for a given task (example parsimony through *structural discriminability*)
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

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 but are very sparse (most but not all entries are zero)

• The “explicit” graph structure approach
  – Graph structure itself represents control sequence mechanism and parameter tying in a statistical model.
Basic Triangle Structures:

A basic explicit approach for parameter tying

Zweig & Russel, ‘99

Counter
Transition
Phone
Observation

- 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.
A Pause For Questions

- Questions?
- Ideas?
- Confusion?
- Suggestions?
- Please ask.
Outline

1. Properties
   a) Overview and Motivation
   b) GM Types and Constructs
   c) Theory and Practice of Dynamic GMs
   d) Explicit Temporal Structures

2. Specific Models
   a) GMs in Speech
   b) GMs in Language
   c) GMs in Machine Translation

3. Toolkits

4. Discussion/Open Questions
Explicit Bi-gram Training Graph Structure

Nodes & Edge Colors:
- Red ⇔ RANDOM
- Green ⇔ Deterministic

- Skip Silence
- Word Counter
- End-of-Utterance Observation=1
- Word
- Word Transition
- State Counter
- State Transition
- State
- Observation

GMs in Audio, Speech, and Language

Jeff A. Bilmes
Explicit Bi-gram Training Graph Structure

Nodes & Edge Colors:
- Red ⇔ RANDOM
- Green ⇔ Deterministic

Skip Silence
Word Counter

Observation = 1
End-of-Utterance

Word
Word Transition
State Counter
State Transition
State
Observation
A Pause For Questions

- Questions?
- Ideas?
- Confusion?
- Suggestions?
- Please ask.
Bi-gram Training w. Pronunciation Variant

Skip Silence

Word Counter

Word

Nodes & Edge Colors:
Red ⇔ RANDOM
Green ⇔ Deterministic

End-of-Utterance Observation=1

Pronunciation

Word Transition

State Counter

State Transition

State

Observation
Switching Parents: Value-specific conditional independence

\[ P(C \mid M_1, F_1, F_2, F_2) = \sum_i P(C \mid M_i, F_i, S \in R_i) P(S \in R_i) \]
Explicit bi-gram Decoder

Word Transition is a switching parent of Word. It switches the implementation of Word(t) to either be a copy of Word(t-1) or to invoke the bi-gram $P(w_t | w_{t-1})$.
Explicit tri-gram Decoder

\[ P(w_t \mid w_{t-1}, w_{t-2}) \]

Previous Word

Nodes & Edges:
- Red ⇔ RANDOM
- Green ⇔ Deterministic
- Dashed line ⇔ Switching Parent

End-of-Utterance Observation=1

Word

Word Transition

State Counter

State Transition

State

Observation
“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

The Hidden Variable Cloud

Say, for this element (suppose we name it $X_{ti}$)

These are the feature elements that comprise $z$

The implementation of these edges determines $f(z)$. Could be linear $Bz$ or non-linear
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.
Multi-stream buried Markov models

\[ Q_{1:T} = q_{1:T} \]

\[ Q_{1:T} = q'_{1:T} \]
Conversational Model

Channel 1

Acoustic Model

Word Transition

Word

Previous Word

End-of-Conv.

Previous Word

Word

Word Transition

Observation

Channel 2

Acoustic Model

Observation
Markov Decision Processes (a digression)

Goal: maximize the sum of rewards, obtainable using normal forward algorithm (dynamic programming)
From Explicit Control to Latent Modeling

1. In latent modeling, we move more towards representing and learning additional information in (factored) hidden space.

2. Factored representations place constraints on what would be flattened HMM transition matrix parameters thereby potentially improving estimation quality.
Latent Modeling

The Hidden Variable Cloud

Observations $X_{1:T}$

• 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

The Hidden Variable Cloud

Observations

Other hidden variables

- 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
Phone-free Articulatory Graph
(by Karen Livescu)
A Pause For Questions

• Questions?
• Ideas?
• Confusion?
• Suggestions?
• Please ask.
Outline

1. Properties
   a) Overview and Motivation
   b) GM Types and Constructs
   c) Theory and Practice of Dynamic GMs
   d) Explicit Temporal Structures

2. Specific Models
   a) GMs in Speech
   b) GMs in Language
   c) GMs in Machine Translation

3. Toolkits

4. Discussion/Open Questions
Part of speech Tagging

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

  ![Diagram of HMMs for word tagging]

- Discriminative models for this task

  ![Diagram of discriminative models for word tagging]

- Label bias issue and selection bias.
Standard Language Modeling

- Example: standard 4-gram

\[
P(w_t | h_t) = P(w_t | w_{t-1}, w_{t-2}, w_{t-3})
\]
Interpolated Uni-, Bi-, Tri-grams

\[ P(w_t \mid h_t) = P(\alpha_t = 1)P(w_t) + P(\alpha_t = 2)P(w_t \mid w_{t-1}) \]

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

- Nothing gets zero probability
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 other graph modules might depend on them (e.g., acoustics, prosody, meaning in silence for MT).
• Goal: allow silence between words, but retain true word predictability skipping silence regions.
• Switching parents can facilitate such a model.
Skip Bi-gram

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

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

\[ p(s_t = 1) = \Pr(\text{silence}) \]
Skip bi-gram with conditional mixtures

\[
p_{\text{bigram}}(w_t \mid r_{t-1}) = P(\alpha_t = 1 \mid w_{t-1})P(w_t) \\
+ P(\alpha_t = 2 \mid w_{t-1})P(w_t \mid r_{t-1})
\]
Two switching parents

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

\[
p(w_t \mid w_{t-1}) = P(\beta_t = 1 \mid w_{t-1})P_{bi}(w_t \mid w_{t-1}) \\
+ P(\beta_t = 0 \mid w_{t-1})P(w_t)
\]
Skip trigram

• Similar to skip bi-gram, but skips over two previous <sil> tokens.
• \( P(\text{City}|<\text{sil}>,\text{York},<\text{sil}>,\text{New}) = P(\text{City}|\text{York},\text{New}) \)
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: \( \Phi = \{ \text{unk} \} \cup \bullet \cup \circ \cup \star \)
- \( \bullet \) = singletons, \( \circ \) = “many-tons”, unk = unknown
- ML distribution gives zero probability to unk.
- Goal: Directed GM that represents and learns \( \alpha \):

\[
p(w) = \begin{cases} 
(1 - \alpha)p_{ml}(\mathcal{S}) & \text{if } w = \text{unk} \\
\alpha p_{ml}(w) & \text{if } w \in \mathcal{S} \\
p_{ml}(w) & \text{otherwise}
\end{cases}
\]

- Word variable is like a switching parent of itself (but of course can’t be, no directed cycles allowed.)
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+unk ($K=1$) vs. “many-ton” ($K=0$)
  – $B =$ indicator of singleton ($B=1$) vs. unknown word ($B=0$)

• Fixed observation child $V$ induces “reverse causal” phenomena via its dependency implementation
  – I.e., child says “if you want me to give you non-zero probability on this observation, you parents had better do $X$”
Explicit Smoothing

\[ P(B = 1) = 1 - P(B = 0) = \alpha \]

\[ P(K = 1) = 1 - P(K = 0) = P(S) \]

\[ p(w \mid k, b) = \begin{cases} 
  p_M(w) & \text{if } k = 0 \\
  p_S(w) & \text{if } k = 1 \text{ and } b = 1 \\
  \delta_{w_1 = \text{unk}} & \text{if } k = 1 \text{ and } b = 0 
\end{cases} \]

\[ P(V = 1 \mid w, k) = 1 \{ (w \in M, k = 0) \text{ or } (w \in S, k = 1) \} \]

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

\[ p_S(w) = \begin{cases} 
  \frac{p_{ml}(w)}{p_{ml}(S)} & \text{if } w \in S \\
  0 & \text{else} 
\end{cases} \]
Putting it together: Class Language Model with smoothing constraints
Factored Language Models (Katrin Kirchhoff)
A Pause For Questions

- Questions?
- Ideas?
- Confusion?
- Suggestions?
- Please ask.
Outline

1. Properties
   a) Overview and Motivation
   b) GM Types and Constructs
   c) Theory and Practice of Dynamic GMs
   d) Explicit Temporal Structures

2. Specific Models
   a) GMs in Speech
   b) GMs in Language
   c) GMs in Machine Translation

3. Toolkits

4. Discussion/Open Questions
Switching Parents

\[ P(C | M1, F1, F2, F2) = \sum_i P(C | M_i, F_i, S = i)P(S \in R_i) \]
“Switching Existence” Variables

- Random variables whose values determine the number of other variables in the graph
- Can represent a random number of random variables
- E.g.: can represent the length of hidden sequences, *fertility* in IBM models 3-5, etc.
Switching Existence: Parameter Tying and Connection to HMMs

- Parameter tying is key for implementing switching existence variables: when a new variable comes into existence, it needs some way to obtain its parameters.

- Time homogeneous HMMs are also implicitly based on parameter tying, i.e., all state variables share the same transition matrix.

- Probabilistic Relational Models also use a similar mechanism: a variable number of instances of an object class share the same parameters.
**Goal:** translate a French string, \( \tilde{f} = f_1^M = f_1 f_2 \ldots f_M \) to an English string \( \tilde{e} = e_1^L \)

**Model:** Noisy channel -- \( \tilde{f} \) is a noisy version of \( \tilde{e} \)

**Inference:** recover most likely \( \tilde{e} \) given \( \tilde{f} \)
IBM Models

- Introduce one-to-many alignment variables $a_{1:M}$ — each French word can translate into at most one English word.

$$p(f, a | e, M) = p(M | e) \prod_{j=1}^{M} p(a_j | a_{1:j-1}^{j-1}, f_{1:j-1}^{j-1}, M, e) p(f_j | a_{1:j}, f_{1:j-1}^{j-1}, M, e)$$  \(1\)
IBM Model 1

- Independence assumptions:
  - \( p(M|\bar{e}) = \text{constant} \)
  - \( p(a_j|a_1^{j-1}, f_1^{j-1}, M, \bar{e}) = p(a_j|L) \)
  - \( p(f_j|a_1^j, f_1^{j-1}, M, \bar{e}) = p(f_j|e_{a_j}) \)

- Switching parents
- Note: parameters for \( a_{1:M} \) are tied (i.e. position independent)
A Pause For Questions

• Questions?
• Ideas?
• Confusion?
• Suggestions?
• Please ask.
Outline

1. Properties
   a) Overview and Motivation
   b) GM Types and Constructs
   c) Theory and Practice of Dynamic GMs
   d) Explicit Temporal Structures

2. Specific Models
   a) GMs in Speech
   b) GMs in Language
   c) GMs in Machine Translation

3. Toolkits

4. Discussion/Open Questions
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.)
- Freely available, to be open-source
GMTK Features

1. Textual Graph Language
2. Switching Parent Functionality
3. Forwards and Backwards time links
4. Multi-rate models with extended DBN templates.
5. Linear Dependencies on observations
6. Arbitrary low-level parameter sharing (EM/GEM training)
8. Decision-Tree-Based implementations of dependencies (deterministic, sparse, formula leaf nodes)
9. Full inference, single pass decoding possible
10. Sampling Methods
11. Linear and Island Algorithm (O(logT)) Exact Inference
GMTK Structure file for HMM

- Structure file defines a prologue \( \mathcal{P} \), chunk \( \mathcal{C} \), and epilog \( \mathcal{E} \). 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

Prologue  Repeating Chunk  Epilogue

Prologue  Chunk Unrolled 1 time  Epilogue
GMTK Structure file for HMM

```plaintext
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");
    }
}
```
Graphical Models

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");
}
Decision-tree implementation of discrete dependencies

\[ X_1 \]
\[ X_2 \]

\[ Q_1(X_1) = T \]
\[ Q_1(X_1) = F \]

\[ Q_2 \]
\[ Q_3 \]
\[ Q_4 \]
\[ Q_5 \]
\[ Q_6 \]
\[ Q_7 \]

\[ P_A(X_2) \]
\[ P_B(X_2) \]
\[ P_C(X_2) \]
\[ P_D(X_2) \]
A Gaussian can be viewed as a directed graphical model.

FSICMs, obtained via U’DU factorization, provides the edge coefficients.

\[
K = U' DU = (I - B)' D(I - B)
\]

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

\[
f(x) = \prod_i f(x_i | x_{\pi_i})
\]
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.
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

\[(\mu^*, D^*, B^*) = \arg\max_{\mu, D, B} Q(\mu, D, B; \mu^o, D^o, B^o)\]
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 ("On Triangulating Dynamic Graphical Models, UAI’2003,”, Bilmes & Bartels).

- Use message passing order in junction tree that respects directed deterministic dependencies when possible (to cut down on state space)
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 need not worry about intricacies of graph triangulation (user concentrates on model).
- Uses a “boundary algorithm” to find chunks of DBN to triangulate (Bilmes & Bartels, UAI’2003).
- Many different triangulation heuristics implemented, all hidden from user (if she so desires).
Sparse-joins in clique structures

- Fast way to do:
  \[ f(a, b, c, d, e) = \phi_1(a, b, c)\phi_2(b, c, d)\phi_3(d, e)\psi(a, b, c, d, e, f) \]

when \( \phi_i(\bigcirc) \) functions very sparse and very large.
Linear and Island Algorithm (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 called the Island Algorithm (Binder et. al. 1997)
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

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

\[
\beta_i(t) = \sum_j \beta_j(t+1) a_{ij} b_j(x_{t+1})
\]
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 quite large, and its exploration has only just started.
Current Status

I. Old version (developed by Jeff Bilmes & Geoff Zweig) available at:
   B. ~100 pages of documentation
   C. Book chapter on use of graphical models for speech and language
   D. JHU’2001 Workshop technical report

II. New Version Running, Much Faster and with many new features. End Summer’04 beta release.
Other GM Toolkits

• Best place to look: 

• GMTK – optimized for speech/language and DBNs. Summer’04 version will be even more so.

• Other tools
  – HTK
  – AT&T Finite State Tools
Discussion

- Questions?
- Ideas?
- Confusion?
- Suggestions?
- Please ask.
Conclusions

- Graphical Models are very flexible!!
- With right toolkit, possible to rapidly build up a novel statistical idea.
- Space of models is still relatively unexplored, young research area for speech/language/NLP.
The End
thank you!
Conditional Independence

- Notation: \( X \perp\!\!\!\!\!\!\!\!\!\perp Y \mid Z \equiv \)
  \[ P(x, y \mid z) = p(x \mid z)p(y \mid z) \quad \forall \{x, y, z\} \]
- 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, \( A' \subseteq A, B' \subseteq B \)
\[ X_A = \{X_{A_1}, X_{A_2}, \ldots, X_{A_N}\} \]