CS 224S/LING 281
Speech Recognition, Synthesis, and Dialogue

Dan Jurafsky

Lecture 14:

Dialogue: MDPs

and

Speaker Detection

Outline for today

- MDP Dialogue Architectures
- Speaker Recognition

Now that we have a success metric

- Could we use it to help drive learning?
- In recent work we use this metric to help us learn an optimal policy or strategy for how the conversational agent should behave

New Idea: Modeling a dialogue system as a probabilistic agent

- A conversational agent can be characterized by:
  - The current knowledge of the system
    - A set of states S the agent can be in
  - a set of actions A the agent can take
  - A goal G, which implies
    - A success metric that tells us how well the agent achieved its goal
    - A way of using this metric to create a strategy or policy π for what action to take in any particular state.

What do we mean by actions A and policies π?

- Kinds of decisions a conversational agent needs to make:
  - When should I ground/confirm/reject/ask for clarification on what the user just said?
  - When should I ask a directive prompt, when an open prompt?
  - When should I use user, system, or mixed initiative?

A threshold is a human-designed policy!

- Could we learn what the right action is
  - Rejection
  - Explicit confirmation
  - Implicit confirmation
  - No confirmation
- By learning a policy which,
  - given various information about the current state,
  - dynamically chooses the action which maximizes dialogue success

Another strategy decision

- Open versus directive prompts
- When to do mixed initiative

- How we do this optimization?
- Markov Decision Processes

Review: Open vs. Directive Prompts

- Open prompt
  - System gives user very few constraints
  - User can respond how they please:
  - "How may I help you?" "How may I direct your call?"
- Directive prompt
  - Explicit instructs user how to respond
  - "Say yes if you accept the call; otherwise, say no"

Review: Restrictive vs. Non-restrictive gramamrs

- Restrictive grammar
  - Language model which strongly constrains the ASR system, based on dialogue state
- Non-restrictive grammar
  - Open language model which is not restricted to a particular dialogue state

Kinds of Initiative

- How do I decide which of these initiatives to use at each point in the dialogue?

Grammar	Open Prompt	Directive Prompt
Restrictive	*Doesn't make sense*	System Initiative
Non-restrictive	User Initiative	Mixed Initiative

Modeling a dialogue system as a probabilistic agent

- A conversational agent can be characterized by:
  - The current knowledge of the system
    - A set of states S the agent can be in
  - a set of actions A the agent can take
  - A goal G, which implies
    - A success metric that tells us how well the agent achieved its goal
    - A way of using this metric to create a strategy or policy π for what action to take in any particular state.

Goals are not enough

- Goal: user satisfaction
- OK, that's all very well, but
  - Many things influence user satisfaction
  - We don't know user satisfaction til after the dialogue is done
  - How do we know, state by state and action by action, what the agent should do?
- We need a more helpful metric that can apply to each state

Utility

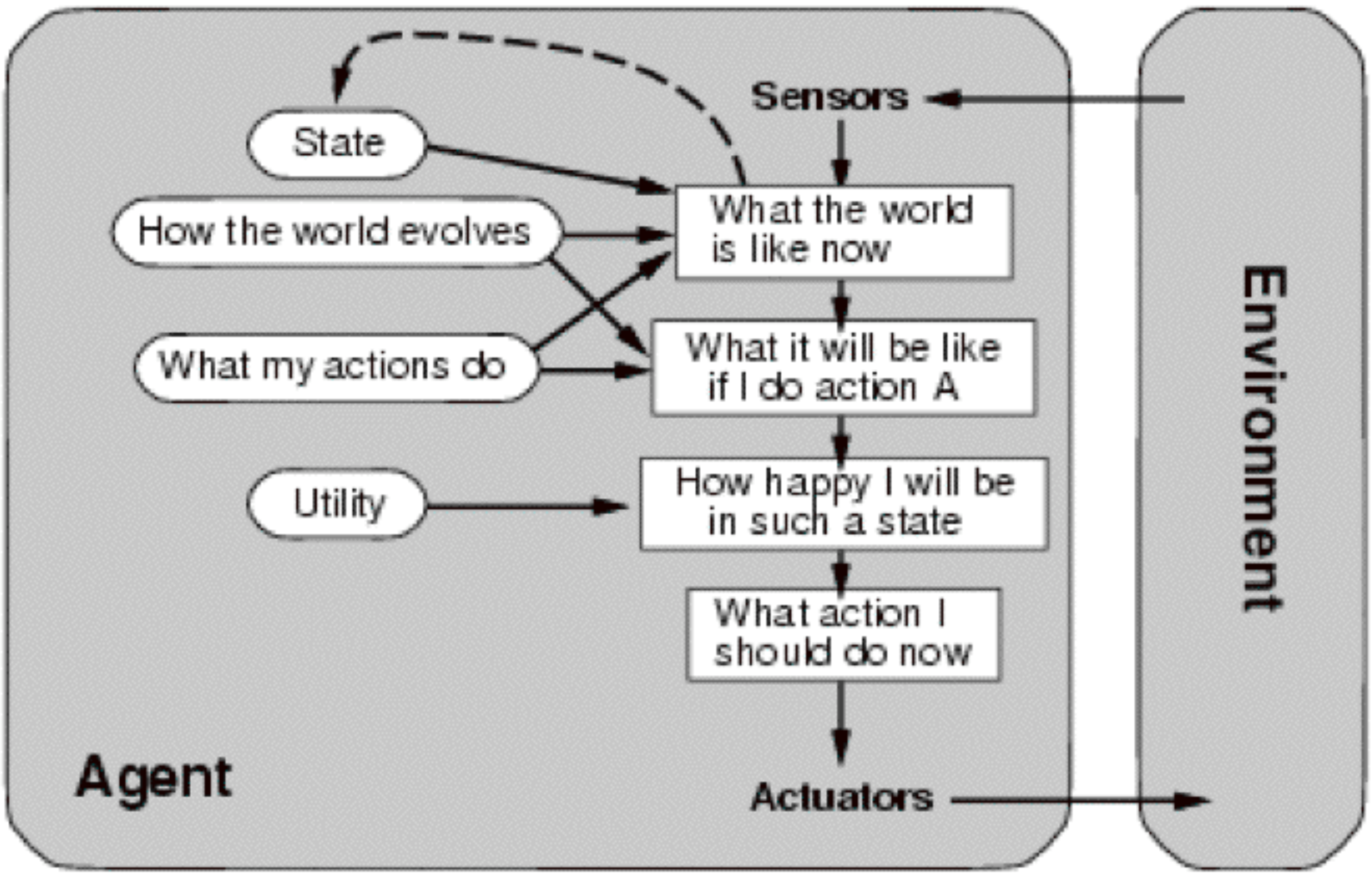
- A utility function
  - maps a state or state sequence
  - onto a real number
  - describing the goodness of that state
  - I.e. the resulting “happiness” of the agent
- Principle of Maximum Expected Utility:
  - A rational agent should choose an action that maximizes the agent’s expected utility

Maximum Expected Utility

- Principle of Maximum Expected Utility:
  - A rational agent should choose an action that maximizes the agent's expected utility
- Action A has possible outcome states $Result_i(A)$
- E: agent's evidence about current state of world
- Before doing A, agent estimates prob of each outcome
  - $P(Result_i(A)|Do(A),E)$
- Thus can compute expected utility:

$$EU(A \mid E) = \sum_i P(Result_i(A) \mid Do(A), E)\, U(Result_i(A))$$

Utility (Russell and Norvig)



Markov Decision Processes

- Or MDP
- Characterized by:
  - a set of states S an agent can be in
  - a set of actions A the agent can take
  - A reward r(a,s) that the agent receives for taking an action in a state
  - (+ Some other things I'll come back to (gamma, state transition probabilities))

A brief tutorial example

- Levin et al (2000)
- A Day-and-Month dialogue system
- Goal: fill in a two-slot frame:
  - Month: November
  - Day: 12th
- Via the shortest possible interaction with user

What is a state?

- In principle, MDP state could include any possible information about dialogue
  - Complete dialogue history so far
- Usually use a much more limited set
  - Values of slots in current frame
  - Most recent question asked to user
  - Users most recent answer
  - ASR confidence
  - etc

State in the Day-and-Month example

- Values of the two slots day and month.
- Total:
  - 2 special initial state s_i and s_f.
  - 365 states with a day and month
  - 1 state for leap year
  - 12 states with a month but no day
  - 31 states with a day but no month
  - 411 total states

Actions in MDP models of dialogue

- Speech acts!
  - Ask a question
  - Explicit confirmation
  - Rejection
  - Give the user some database information
  - Tell the user their choices
- Do a database query

Actions in the Day-and-Month example

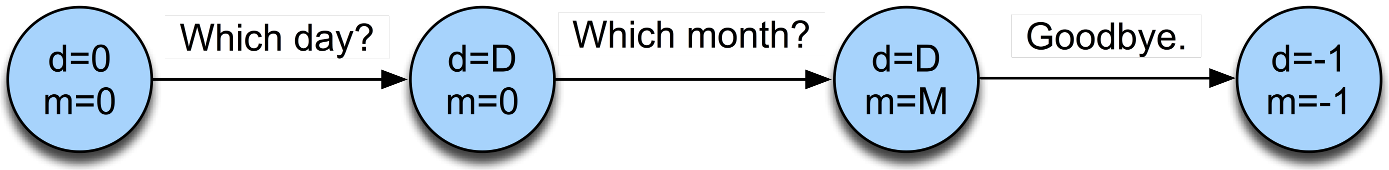
- a_d: a question asking for the day
- a_m: a question asking for the month
- a_{dm}: a question asking for the day+month
- a_f: a final action submitting the form and terminating the dialogue

A simple reward function

- For this example, let's use a cost function
- A cost function for entire dialogue
- Let
  - N_i=number of interactions (duration of dialogue)
  - N_e=number of errors in the obtained values (0-2)
  - N_f=expected distance from goal
    - (0 for complete date, 1 if either data or month are missing, 2 if both missing)
- Then (weighted) cost is:
- $C = w_i \times N_i + w_e \times N_e + w_f \times N_f$

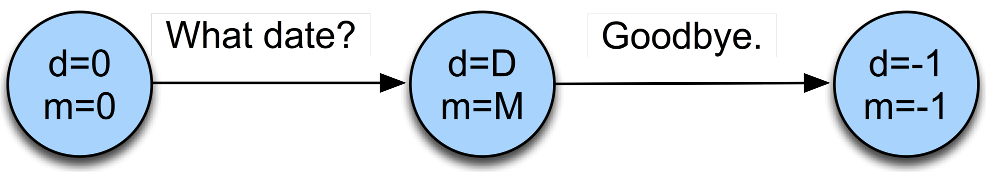
2 possible policies

Policy 1 (directive)



$c_1 = -3w_i + 2p_d w_e$

Policy 2 (open)



$c_2 = -2w_i + 2p_o w_e$

P_d=probability of error in directive prompt

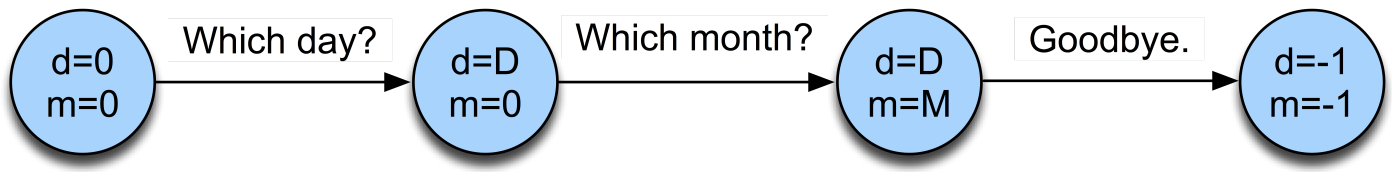
P_o=probability of error in open prompt

2 possible policies

Strategy 1 is better than strategy 2 when improved error rate justifies longer interaction:

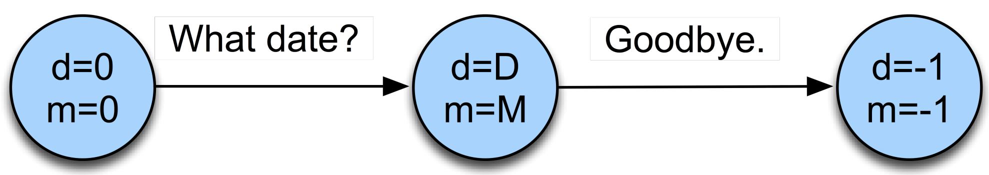
$$p_o - p_d > \frac{w_i}{2w_e}$$

Policy 1 (directive)



$c_1 = -3w_i + 2p_d w_e$

Policy 2 (open)



$c_2 = -2w_i + 2p_o w_e$

That was an easy optimization

- Only two actions, only tiny # of policies
- In general, number of actions, states, policies is quite large
- So finding optimal policy π^* is harder
- We need reinforcement leraning
- Back to MDPs:

MDP

- We can think of a dialogue as a trajectory in state space

- The best policy π^* is the one with the greatest expected reward over all trajectories
- How to compute a reward for a state sequence?

$$s_1 \rightarrow_{a1,r1} s_2 \rightarrow_{a2,r2} s_3 \rightarrow_{a3,r3} \cdots$$

Reward for a state sequence

- One common approach: discounted rewards
- Cumulative reward Q of a sequence is discounted sum of utilities of individual states

$$Q([s_0, a_0, s_1, a_1, s_2, a_2 \cdots]) = R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots ,$$

- Discount factor γ between 0 and 1
- Makes agent care more about current than future rewards; the more future a reward, the more discounted its value

The Markov assumption

- MDP assumes that state transitions are Markovian

$$P(s_{t+1} \mid s_t, s_{t-1}, \ldots, s_o, a_t, a_{t-1}, \ldots, a_o) = P_T(s_{t+1} \mid s_t, a_t)$$

Expected reward for an action

- Expected cumulative reward Q(s,a) for taking a particular action from a particular state can be computed by Bellman equation:

$$Q(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

- Expected cumulative reward for a given state/ action pair is:
  - immediate reward for current state
  - + expected discounted utility of all possible next states s’
  - Weighted by probability of moving to that state s’
  - And assuming once there we take optimal action a’

What we need for Bellman equation

- A model of p(s'|s,a)
- Estimate of R(s,a)
- How to get these?
- If we had labeled training data
  - P(s'|s,a) = C(s,s',a)/C(s,a)
- If we knew the final reward for whole dialogue R(s1,a1,s2,a2,…,sn)
- Given these parameters, can use value iteration algorithm to learn Q values (pushing back reward values over state sequences) and hence best policy

Final reward

- What is the final reward for whole dialogue R(s1,a1,s2,a2,…,sn)?
- This is what our automatic evaluation metric PARADISE computes!
- The general goodness of a whole dialogue!!!!!

How to estimate p(s'|s,a) without labeled data

- Have random conversations with real people
  - Carefully hand-tune small number of states and policies
  - Then can build a dialogue system which explores state space by generating a few hundred random conversations with real humans
  - Set probabilities from this corpus
- Have random conversations with simulated people
  - Now you can have millions of conversations with simulated people
  - So you can have a slightly larger state space

An example

- Singh, S., D. Litman, M. Kearns, and M. Walker. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. Journal of AI Research.

- NJFun system, people asked questions about recreational activities in New Jersey
- Idea of paper: use reinforcement learning to make a small set of optimal policy decisions

Very small # of states and acts

- **States**: specified by values of 8 features
  - Which slot in frame is being worked on (1-4)
  - ASR confidence value (0-5)
  - How many times a current slot question had been asked
  - Restrictive vs. non-restrictive grammar
  - Result: 62 states
- **Actions**: each state only 2 possible actions
  - Asking questions: System versus user initiative
  - Receiving answers: explicit versus no confirmation.

Ran system with real users

- 311 conversations
- Simple binary reward function
  - 1 if competed task (finding museums, theater, winetasting in NJ area)
  - 0 if not
- System learned good dialogue strategy: Roughly
  - Start with user initiative
  - Backoff to mixed or system initiative when re-asking for an attribute
  - Confirm only a lower confidence values

State of the art

- Only a few such systems
  - From (former) ATT Laboratories researchers, now dispersed
  - And Cambridge UK lab
- Hot topics:
  - Partially observable MDPs (POMDPs)
  - We don't REALLY know the user's state (we only know what we THOUGHT the user said)
  - So need to take actions based on our BELIEF , I.e. a probability distribution over states rather than the "true state"

Summary

- Utility-based conversational agents
  - Policy/strategy for:
    - Confirmation
    - Rejection
    - Open/directive prompts
    - Initiative
    - +?????
  - MDP
  - POMDP

Summary

- The Linguistics of Conversation
- Basic Conversational Agents
  - ASR
  - NLU
  - Generation
  - Dialogue Manager
- Dialogue Manager Design
  - Finite State
  - Frame-based
  - Initiative: User, System, Mixed
- VoiceXML
- Information-State
  - Dialogue-Act Detection
  - Dialogue-Act Generation
- Evaluation
- Utility-based conversational agents
  - MDP, POMDP

Part II: Speaker Recognition

Speaker Recognition tasks

- Speaker Recognition
  - Speaker Verification (Speaker Detection)
    - Is this speech sample from a particular speaker

      Is that Jane?
  - Speaker Identification
    - Which of this set of speakers does this speech sample come from

      Who is that?
    - Related tasks: Gender ID, Language ID

      Is this a woman or a man?
- Speaker Diarization
  - Segmenting a dialogue or multiparty conversation

    Who spoke when?

Speaker Recognition tasks

- Two Modes of Speaker Verification
  - Text-dependent (Text-constrained)
    - There is some constraint on the type of utterance that users of the system can pronounce
  - Text-independent
    - Users can say whatever they want

Slide adapted from Chu, Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-Garcia, Petrovska-Delacretaz, Reynolds

Introduction (cont.)

- Two Cases of Speaker Identification
  - Closed Set
    - A reference model for the unknown speaker may not exist
  - Open Set
    - An additional decision alternative, “the unknown does not match any of the models”, is required

Slide adapted from Chu, Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-Garcia, Petrovska-Delacretaz, Reynolds

Speaker Verification

- Basic idea: likelihood ratio detection
  - Assumption: A segment of speech Y contains speech from only one speaker
  - Hypothesis test:

    H0: Y is from the hypothesized speaker S

    H1: Y is not from the hypothesized speaker S
  - A likelihood ratio (LR) test given by

$$\frac{p(Y|H0)}{p(Y|H1)} \begin{cases} > \Theta, \text{ accept H0,} \\ < \Theta, \text{ accept H1,} \end{cases}$$

Slide adapted from Chu, Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-Garcia, Petrovska-Delacretaz, Reynolds

Speaker ID

Log-Likelihood Ratio Score

- We determine which hypothesis is true using the ratio:

$$\frac{p(X \mid H_0)}{p(X \mid H_1)} \begin{cases} \geq \text{threshold}, & \text{accept } H_0 \\ \leq \text{threshold}, & \text{reject } H_0 \end{cases}$$

$$\Lambda(X) = \log[p(X \mid \lambda_C)] - \log[p(X \mid \lambda_{\overline{C}})]$$

$$\Lambda(X) \begin{cases} \geq \text{threshold}, & X \text{ generated by } \lambda_C \\ < \text{threshold}, & X \text{ generated by } \lambda_{\overline{C}} \end{cases}$$

- We use the *log-likelihood ratio score* to decide whether an observed speaker, language, or dialect is the target

Statistical Modeling (cont.)

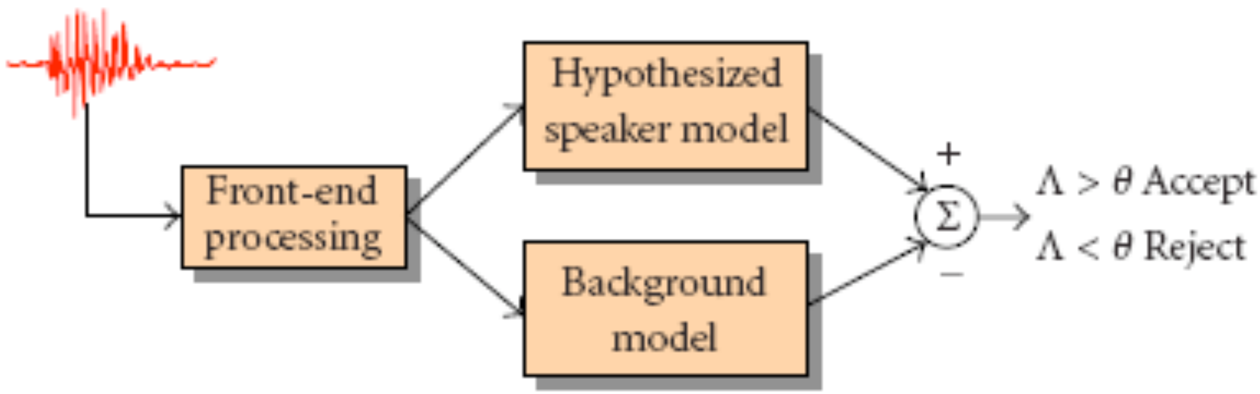


FIGURE 5: Likelihood-ratio-based speaker verification system.

Slide adapted from Chu, Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-Garcia, Petrovska-Delacretaz, Reynolds

How do we get H1?

- Pool speech from several speakers and train a single model:
  - a universal background model (UBM)

- Main advantage :
  - a single speaker-independent model (λ_{bkg}) can be trained once for a particular task and then used for all hypothesized speakers in that task

Slide adapted from Chu, Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-Garcia, Petrovska-Delacretaz, Reynolds

How to compute P(H|X)?

- For text-independent speaker recognition, the most successful likelihood function has been GMMs

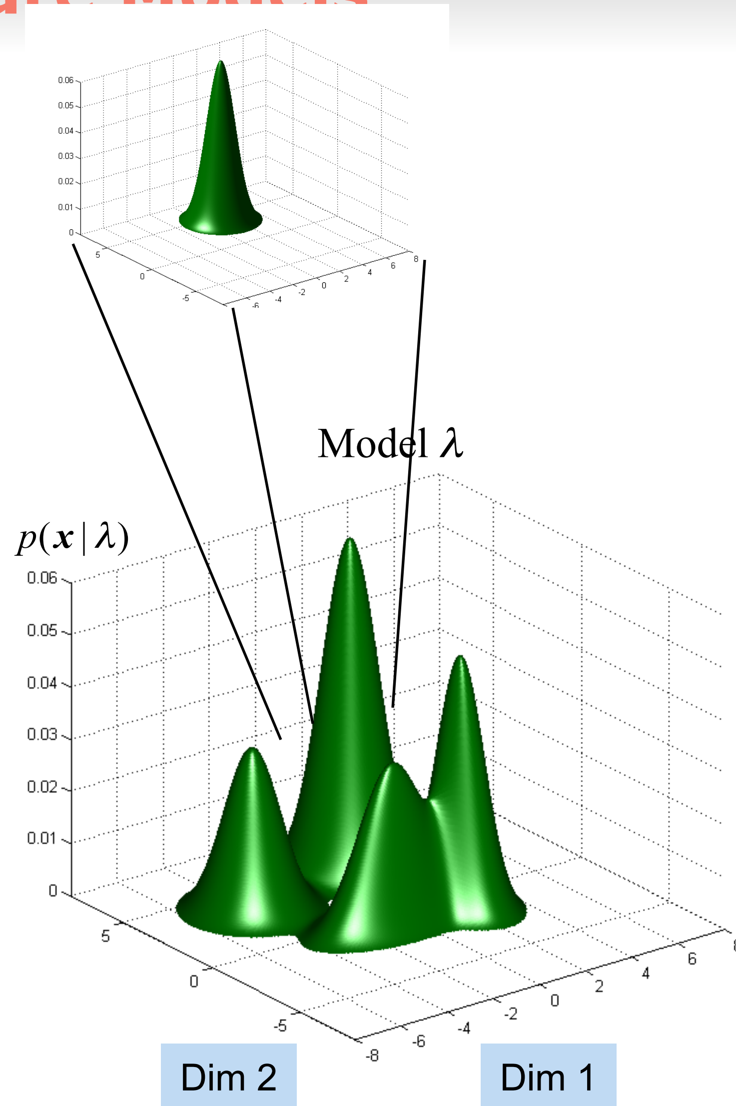
Slide adapted from Chu, Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-Garcia, Petrovska-Delacretaz, Reynolds

Recognition Systems

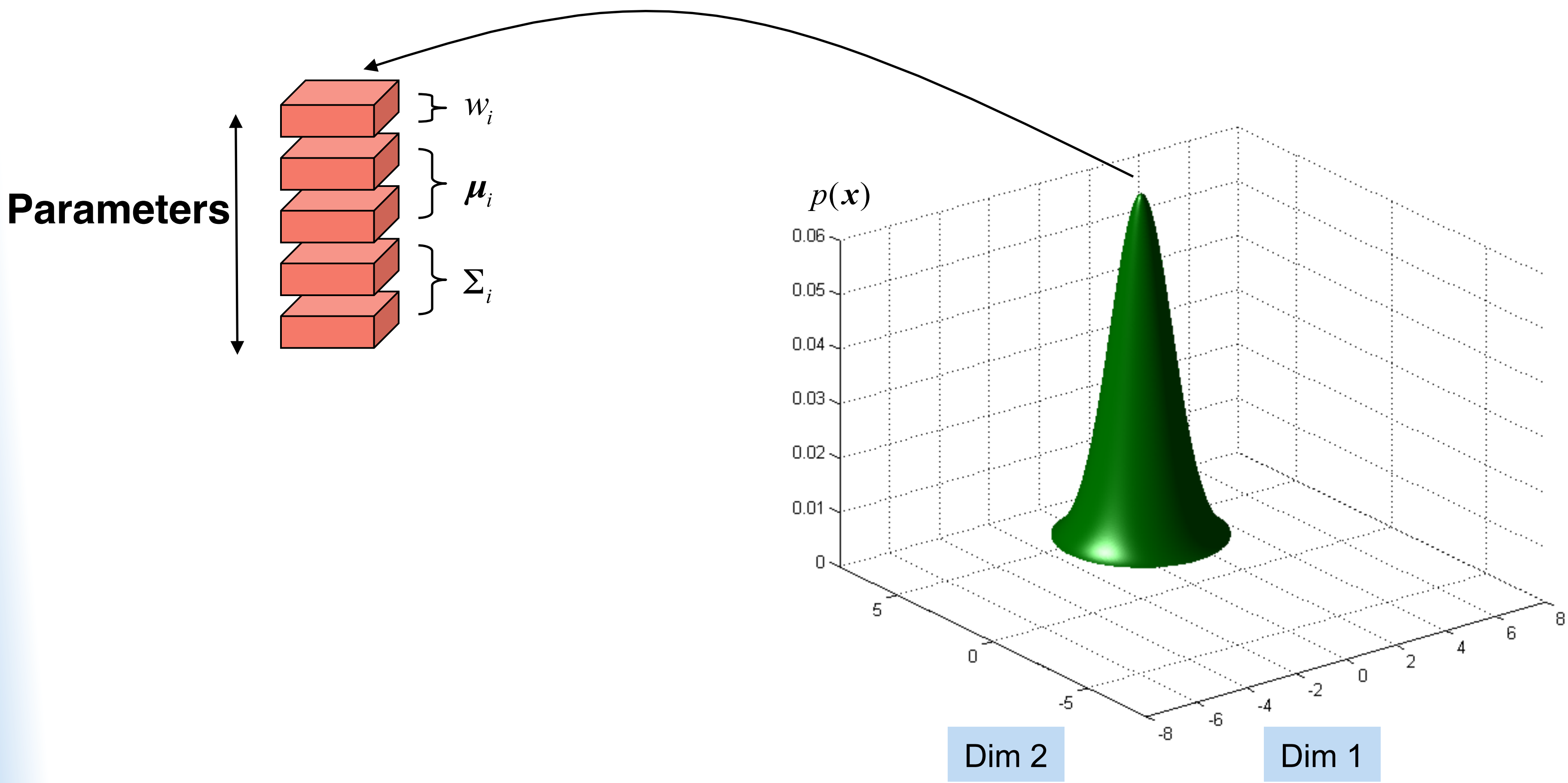
Gaussian Mixture Models

- A Gaussian mixture model (GMM) represents features as the weighted sum of multiple Gaussian distributions

- Each Gaussian *state i* has a
  - Mean $\boldsymbol{\mu}_i$
  - Covariance Σ_i
  - Weight w_i



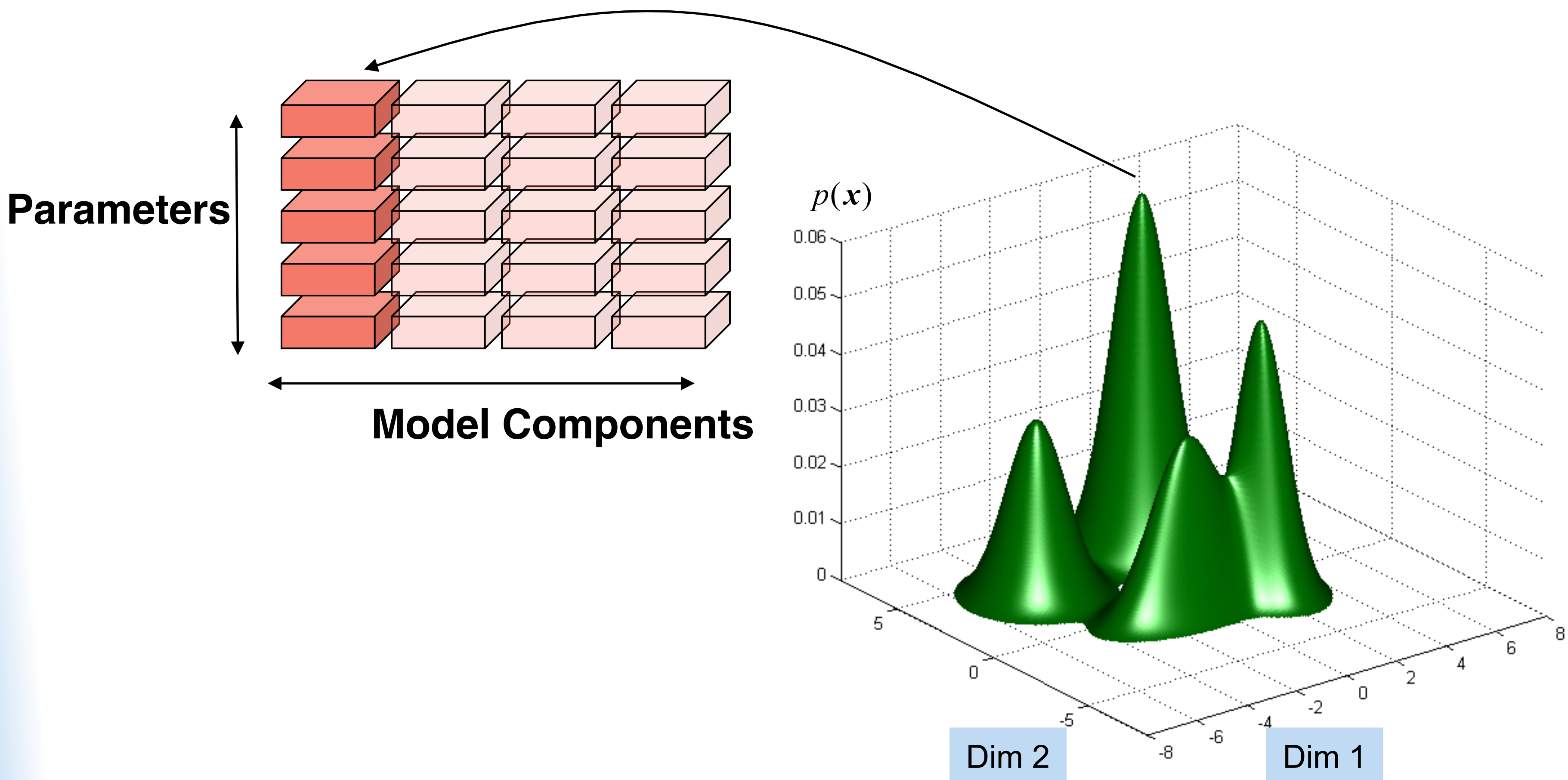
Recognition Systems
Gaussian Mixture Models



Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

Recognition Systems

Gaussian Mixture Models

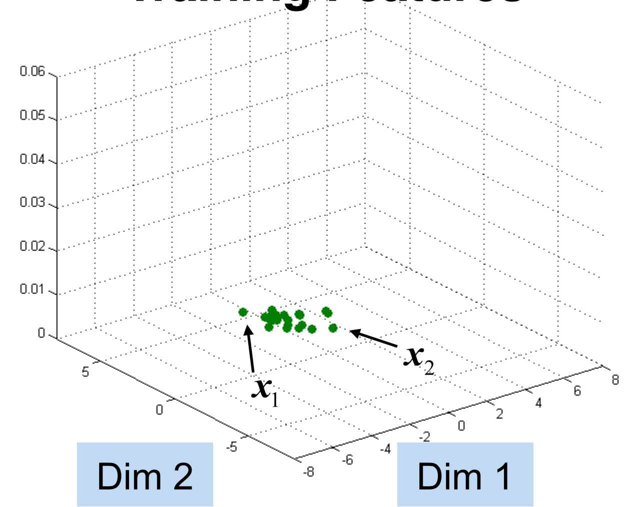


Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

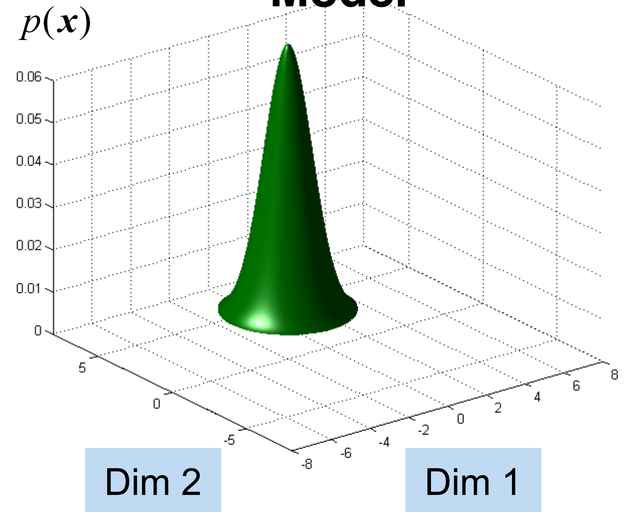
GMM training

- A recognition system makes decisions about observed data based on a knowledge of past data
- During *training*, the system learns about the data it uses to make decisions
  - A set of features are collected from a certain language, dialect, or speaker
  - A *model* is generated to represent the data

Training Features



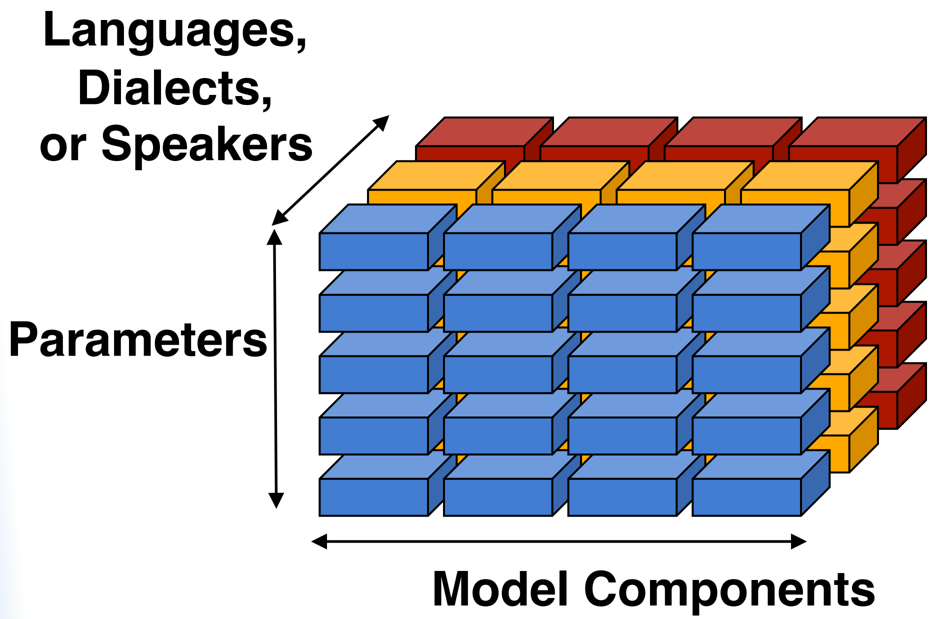
Model

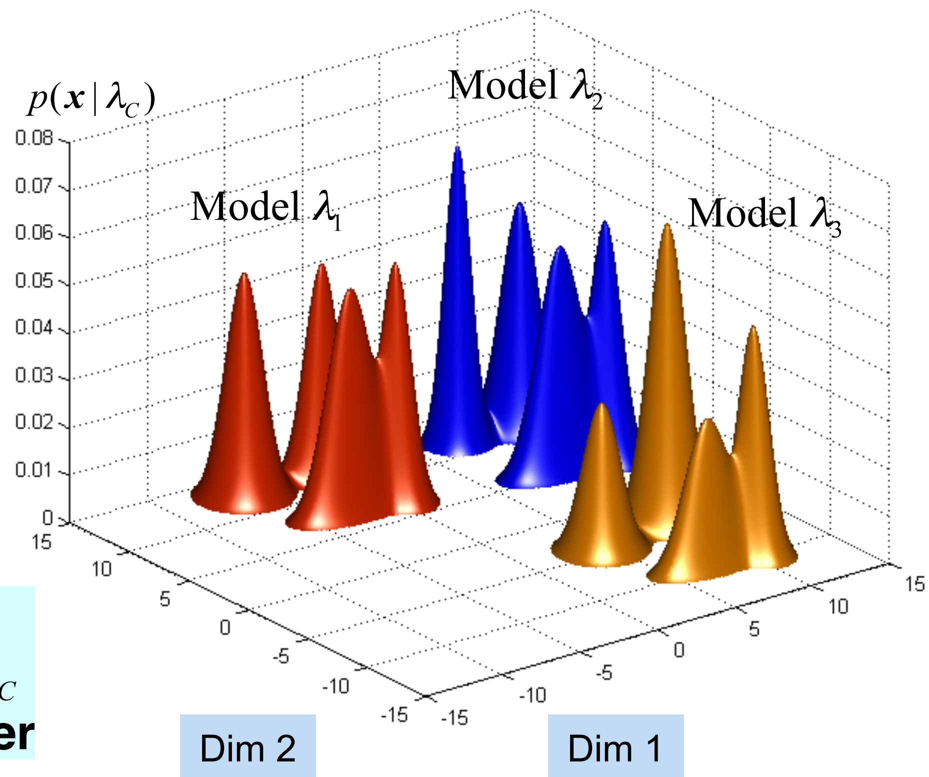


Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

Recognition Systems

Language, Speaker, and Dialect Models

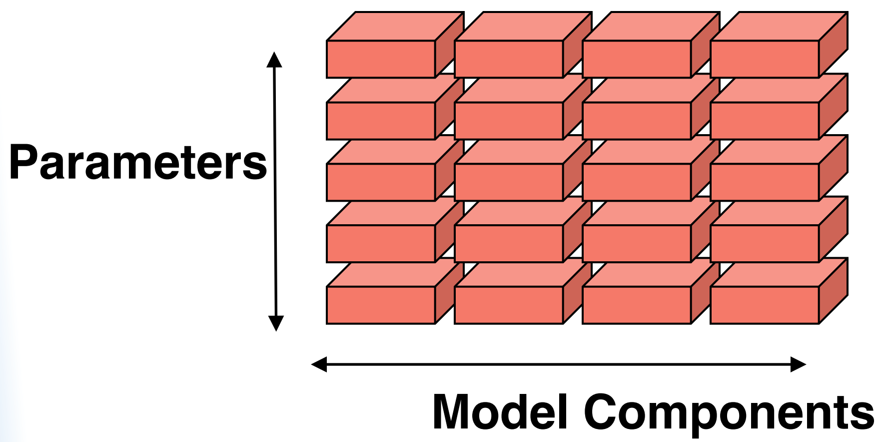


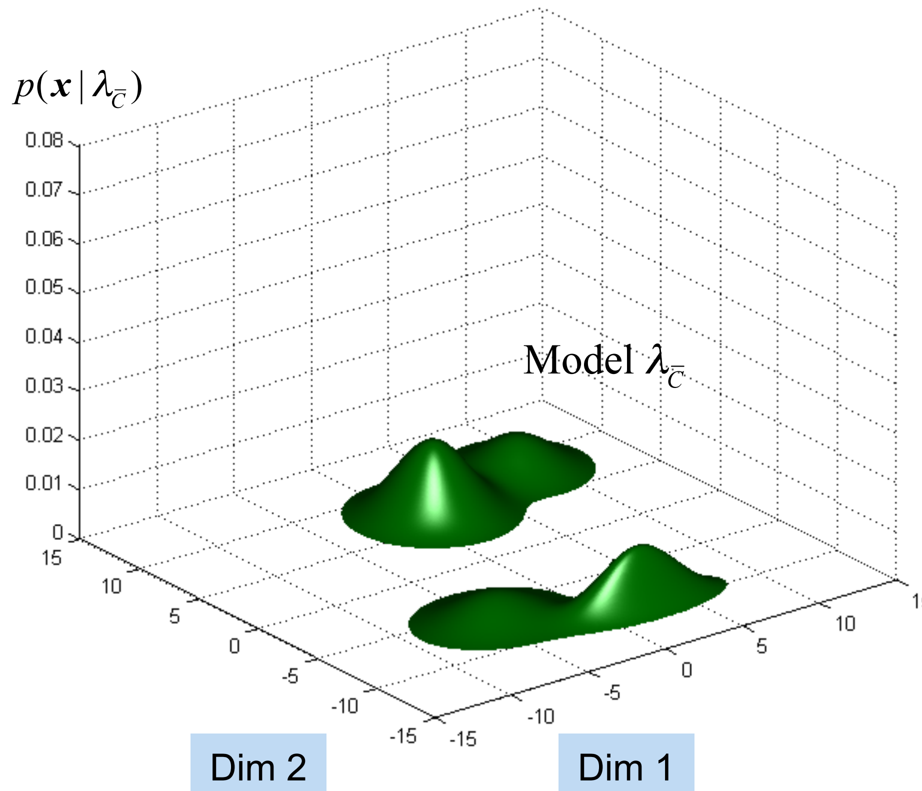


In LID, DID, and SID, we train a set of *target models* λ_C for each dialect, language, or speaker

Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

Recognition Systems
Universal Background Model





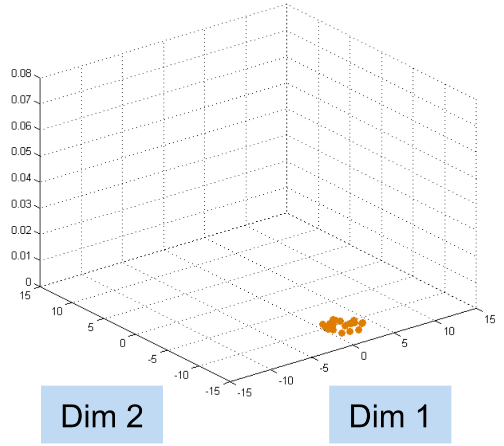
We also train a *universal background model* $\lambda_{\bar{C}}$ representing all speech

Recognition Systems
Hypothesis Test

- Given a set of *test observations*, we perform a hypothesis test to determine whether a certain class produced it

$$H_0: \quad X_{test} \text{ is from the hypothesized class}$$
$$H_1: \quad X_{test} \text{ is not from the hypothesized class}$$

$$X_{test} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_K\}$$



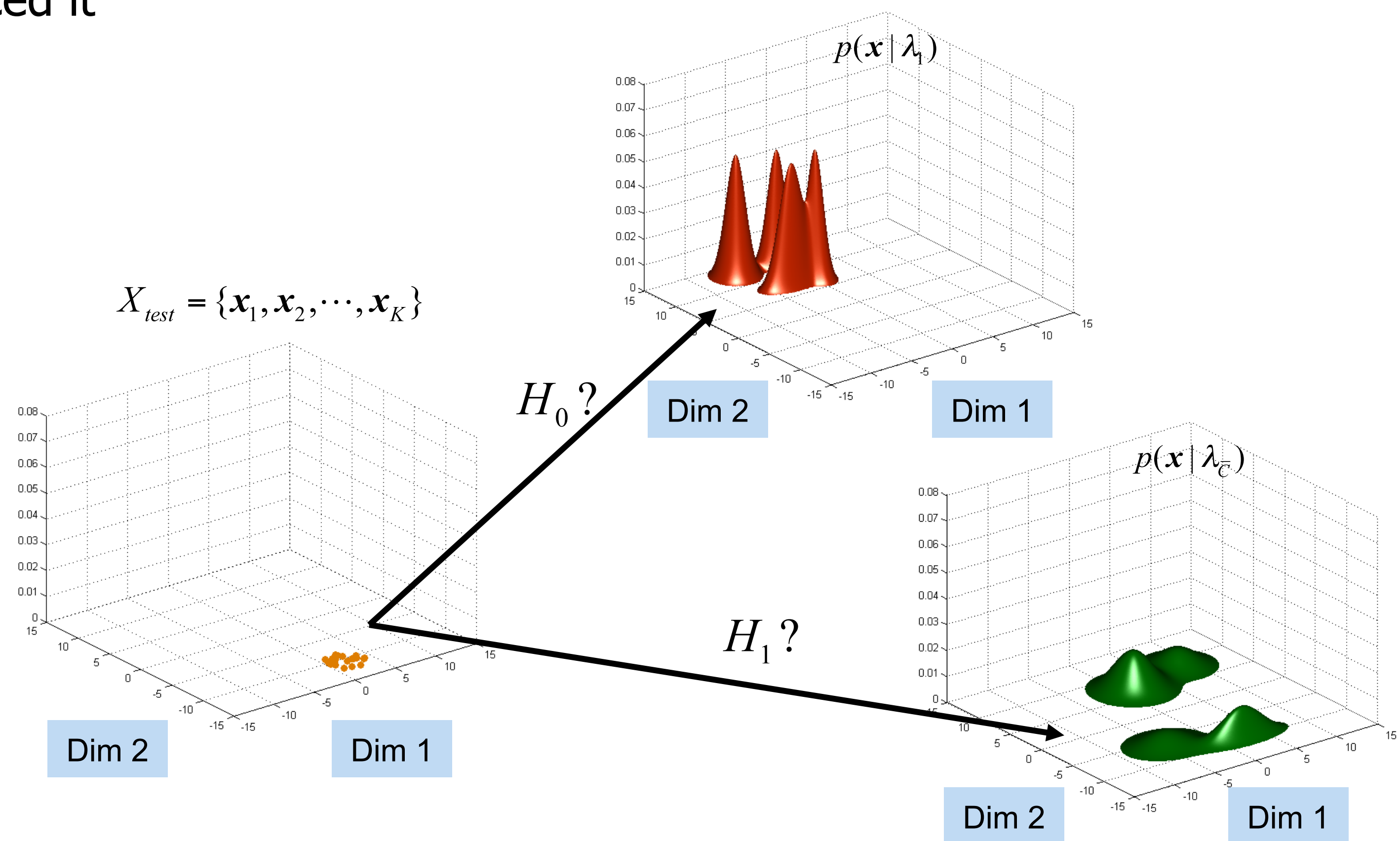
Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

Recognition Systems

Hypothesis Test

- Given a set of *test observations*, we perform a hypothesis test to determine whether a certain class produced it

$$H_0: \quad X_{test} \text{ is from the hypothesized class}$$
$$H_1: \quad X_{test} \text{ is not from the hypothesized class}$$

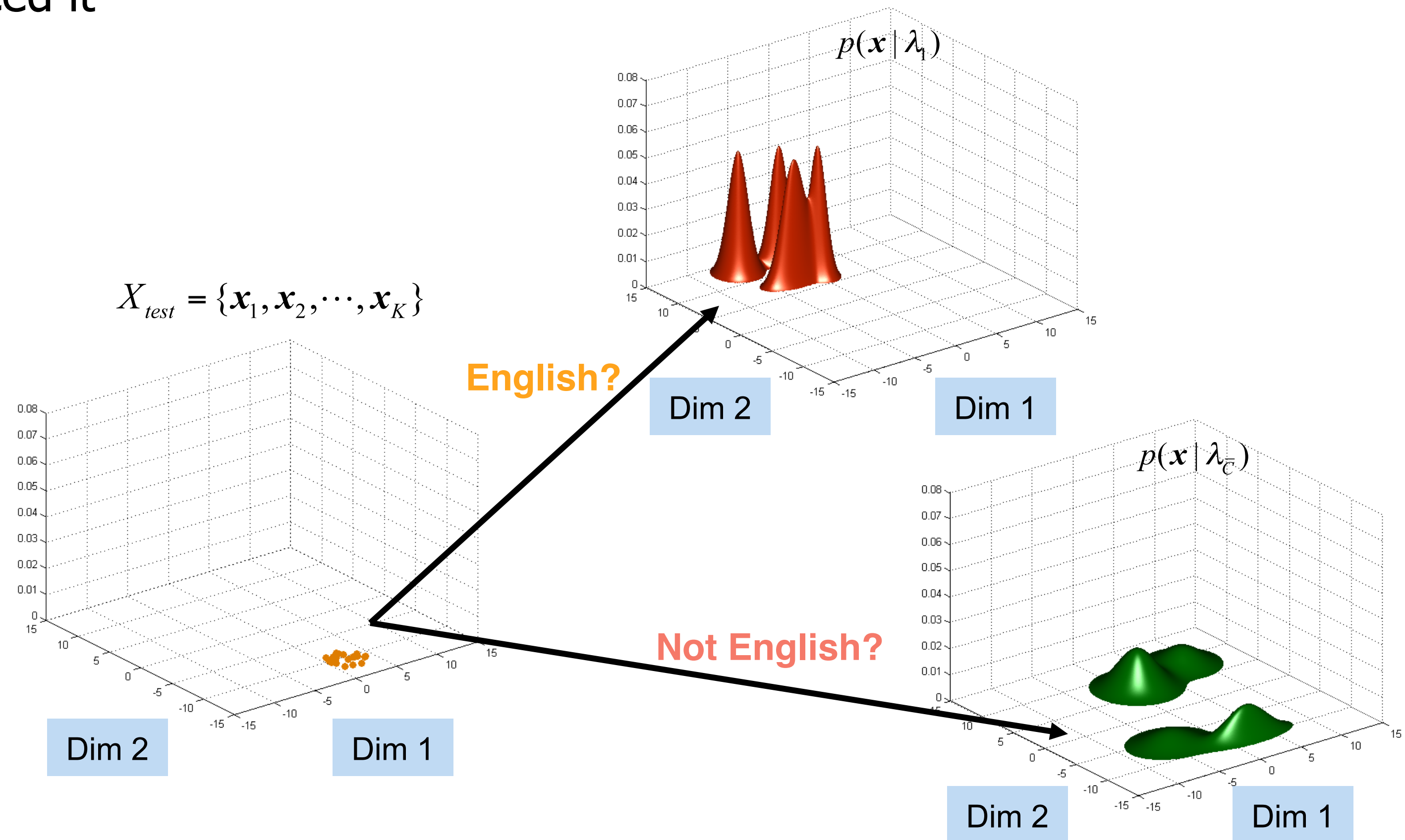


Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

Recognition Systems

Hypothesis Test

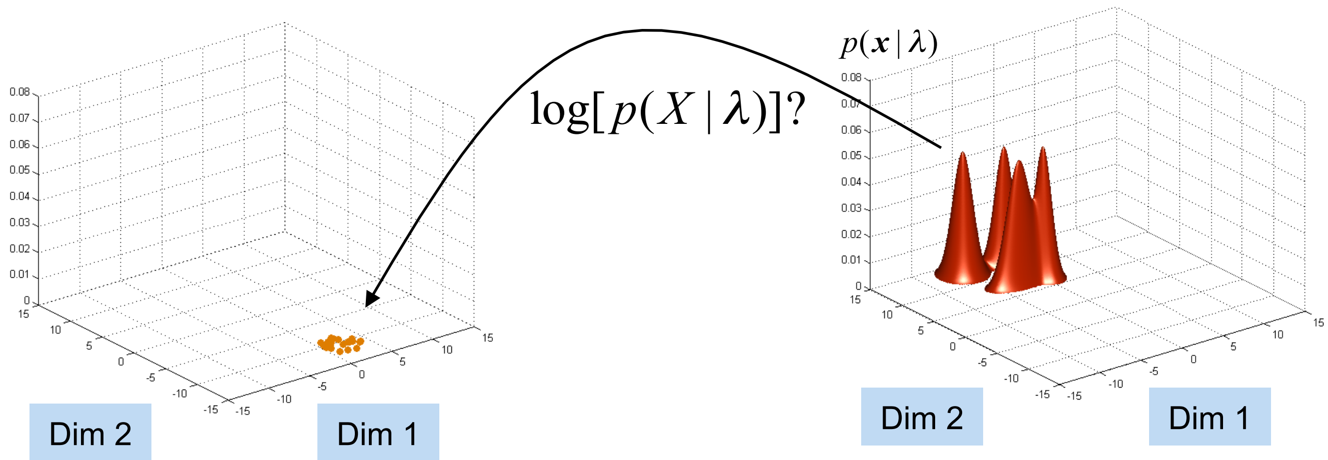
- Given a set of *test observations*, we perform a hypothesis test to determine whether a certain class produced it



Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

Recognition Systems
Log-Likelihood Computation

- The observation log-likelihood given a model λ is:



Nicolas Malyska, Sanjeev Mohindra, Karen Lauro, Douglas Reynolds, and Jeremy Kepner

Gaussian mixture models

- For a *D*-dimensional feature vector $\vec{x}$, the mixture density used for the likelihood function is defined as follows:

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^{M} w_i p_i(\vec{x}) \qquad \sum_{i=1}^{M} w_i = 1,$$

- Gaussian densities $p_i(\vec{x})$, each parameterized by a *D* × 1 mean vector $\vec{u}_i$ and a *D* × *D* covariance matrix Σ *i*:

$$p_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} \mid \Sigma_i \mid^{1/2}} e^{-(1/2)(\vec{x}-\vec{\mu}_i)' \sum_i^{-1} (\vec{x}-\vec{\mu}_i)}$$

- Collectively, the parameters of the density model are denoted as $\lambda = (w_i, \vec{u}_i, \Sigma_i)$, *i* = (1, . . . ,*M*)

Slide adapted from Chu, Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-Garcia, Petrovska-Delacretaz, Reynolds

Gaussian mixture models

- Under the assumption of independent feature vectors, the log-likelihood of a model λ for a sequence of feature vectors $X = \{\vec{x}_1, \ldots, \vec{x}_T\}$ is computed as follows:

$$\log p(X \mid \lambda) = \frac{1}{T} \sum_t \log p(\vec{x}_t \mid \lambda)$$

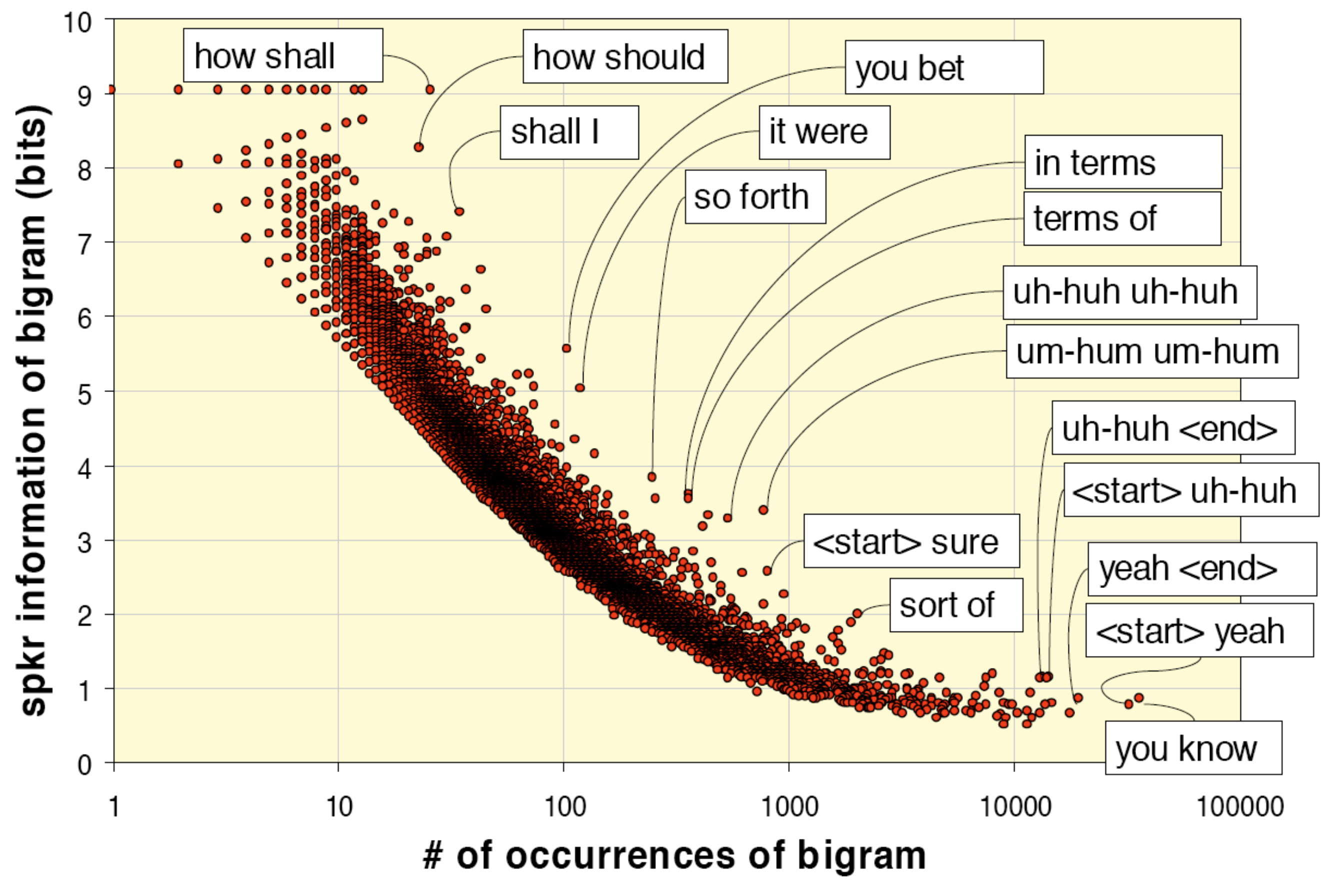
- GMMs are comptuationally inexpensive
  - For homework: single gaussian.
  - Real systems:
    - UBM background model: 512–2048 mixtures
    - Speaker's GMM: 64–256 mixtures

- Recent work:
  - Combining high-level information (such as speaker-dependent word usage or speaking style) with GMMs

Slide adapted from Chu, Bimbot, Bonastre, Fredouille, Gravier, Magrin-Chagnolleau, Meignier, Merlin, Ortega-Garcia, Petrovska-Delacretaz, Reynolds

Doddington (2001)

- Word bigrams can be very informative about speaker identity

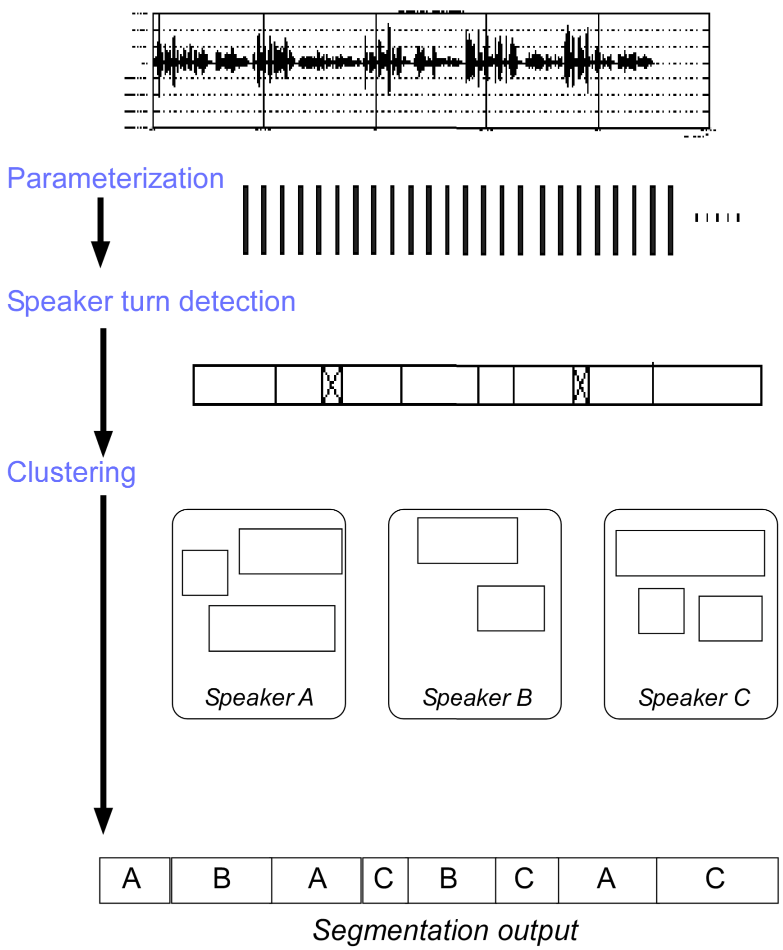


Speaker diarization

- Tasks
  - Conversational telephone speech
    - 2 speakers
  - Broadcast news
    - Many speakers although often in dialogue (interviews) or in sequence (broadcast segments)
  - Meeting recordings
    - Many speakers, lots of overlap and disfluencies
- General 2-step algorithm
  - Segmentation into speakers
    - Detection of speaker-change (insert boundaries)
  - Clustering (MFCC)s of segments

Speaker diarization

- General 2-step algorithm
  - Segmentation into speakers
    - Detection of speaker-change (insert boundaries)
  - Clustering (MFCC)s of segments



Picture from slide by Moraru, Besacier, Meignier, Fredouille, Bonastre

Outline for today

- MDP Dialogue Architectures
- Speaker Recognition