Latent Variable Models
Probabilistic Models in the Study of Language
Day 4

Roger Levy

UC San Diego
Department of Linguistics
Here is the kind of hierarchical model we’ve seen so far:
Plate notation for graphical models

Here is a more succinct representation of the same model:

The rectangles with $N$ and $m$ are plates; semantics of a plate with $n$ is “replicate this node $n$ times”
Plate notation for graphical models

Here is a more succinct representation of the same model:

- The rectangles with $N$ and $m$ are plates; semantics of a plate with $n$ is “replicate this node $n$ times”
- $N = \sum_{i=1}^{m} n_i$ (see previous slide)
Plate notation for graphical models

Here is a more succinct representation of the same model:

- $\theta$
- $y$
- $b$
- $\Sigma$
- $i$
- $m$
- $N$

The rectangles with $N$ and $m$ are plates; semantics of a plate with $n$ is “replicate this node $n$ times”

- $N = \sum_{i=1}^{m} n_i$ (see previous slide)
- The $i$ node is a cluster identity node
Plate notation for graphical models

Here is a more succinct representation of the same model:

The rectangles with $N$ and $m$ are plates; semantics of a plate with $n$ is “replicate this node $n$ times”

$N = \sum_{i=1}^{m} n_i$ (see previous slide)

The $i$ node is a cluster identity node

In our previous application of hierarchical models to regression, cluster identities are known
Plate notation for graphical models

Here is a more succinct representation of the same model:

- The rectangles with $N$ and $m$ are plates; semantics of a plate with $n$ is “replicate this node $n$ times”
- $N = \sum_{i=1}^{m} n_i$ (see previous slide)
- The $i$ node is a cluster identity node
- In our previous application of hierarchical models to regression, cluster identities are known
The plan for today’s lecture

We are going to study the simplest type of latent-variable models.
The plan for today’s lecture

- We are going to study the simplest type of latent-variable models
- Technically speaking, “latent variable” means any variable whose value is unknown
We are going to study the simplest type of latent-variable models.

Technically speaking, “latent variable” means any variable whose value is unknown.

But it’s conventionally used to refer to *hidden structural relations among observations*.
We are going to study the simplest type of latent-variable models.

Technically speaking, “latent variable” means *any* variable whose value is unknown.

But it’s conventionally used to refer to *hidden structural relations among observations*.

In today’s clustering applications, simply treat \( i \) as unknown.
We are going to study the simplest type of latent-variable models.

Technically speaking, “latent variable” means any variable whose value is unknown.

But it’s conventionally used to refer to hidden structural relations among observations.

In today’s clustering applications, simply treat $i$ as unknown.

Inferring values of $i$ induces a clustering among observations; to do so we need to put a probability distribution over $i$. 
The plan for today’s lecture

We will cover two types of simple latent-variable models:
The plan for today’s lecture

We will cover two types of simple latent-variable models:

- The mixture of Gaussians for continuous multivariate data;
The plan for today’s lecture

We will cover two types of simple latent-variable models:
- The mixture of Gaussians for continuous multivariate data;
- Latent Dirichlet Allocation (LDA; also called Topic models) for categorical data (words) in collections of documents.
Mixture of Gaussians

- Motivating example: how are phonological categories learned
Mixture of Gaussians

- Motivating example: how are phonological categories learned
- Evidence that learning involves a combination of both innate bias and experience:
Mixture of Gaussians

- Motivating example: how are phonological categories learned
- Evidence that learning involves a combination of both innate bias and experience:
  - Infants can distinguish some contrasts that adults of speakers lacking them cannot: alveolar [d] versus retroflex [r] for English speakers, [r] versus [l] for Japanese speakers; Werker and Tees, 1984; Kuhl et al., 2006, inter alia)
Motivating example: how are phonological categories learned

Evidence that learning involves a combination of both innate bias and experience:

- Infants can distinguish some contrasts that adults of speakers lacking them cannot: alveolar [d] versus retroflex [ɾ] for English speakers, [ɾ] versus [l] for Japanese speakers; Werker and Tees, 1984; Kuhl et al., 2006, inter alia)
- Other contrasts are not reliably distinguished until ~ 1 year of age by native speakers (e.g., syllable-initial [n] versus [ŋ] in Filipino language environments; Narayan et al., 2010)
Learning vowel categories

To appreciate the potential difficulties of vowel category learning, consider inter-speaker variation (data courtesy of Vallabha et al., 2007):
Framing the category learning problem

Here’s 19 speakers’ data mixed together:
Framing the category learning problem

- “Learning” from such data can be thought of in two ways:
Framing the category learning problem

▶ “Learning” from such data can be thought of in two ways:
  ▶ Grouping the observations into categories
Framing the category learning problem

“Learning” from such data can be thought of in two ways:
- Grouping the observations into categories
- Determining the underlying category representations (positions, shapes, and sizes)
Framing the category learning problem

- “Learning” from such data can be thought of in two ways:
  - Grouping the observations into categories
  - Determining the underlying category representations (positions, shapes, and sizes)
- Formally: every possible grouping of observations $y$ into categories represents a partition $\Pi$ of the observations $y$. 
Framing the category learning problem

“Learning” from such data can be thought of in two ways:

- Grouping the observations into categories
- Determining the underlying category representations (positions, shapes, and sizes)

Formally: every possible grouping of observations \( y \) into categories represents a partition \( \Pi \) of the observations \( y \).
Framing the category learning problem

“Learning” from such data can be thought of in two ways:
• Grouping the observations into categories
• Determining the underlying category representations (positions, shapes, and sizes)

Formally: every possible grouping of observations \( y \) into categories represents a partition \( \Pi \) of the observations \( y \). If \( \theta \) are parameters describing category representations, our problem is to infer

\[
P(\Pi, \theta | y)
\]
Framing the category learning problem

- “Learning” from such data can be thought of in two ways:
  - Grouping the observations into categories
  - Determining the underlying category representations (positions, shapes, and sizes)

- Formally: every possible grouping of observations \( y \) into categories represents a partition \( \Pi \) of the observations \( y \). If \( \theta \) are parameters describing category representations, our problem is to infer

\[
P(\Pi, \theta | y)
\]

from which we could recover the two marginal probability distributions of interest:

\[
P(\Pi | y) \quad \text{(distr. over partitions given data)}
\]

\[
P(\theta | y) \quad \text{(distr. over category properties given data)}
\]
The mixture of Gaussians

- Simple generative model of the data: we have $k$ multivariate Gaussians with frequencies $\phi = \langle \phi_1, \ldots, \phi_k \rangle$, each with its own mean $\mu_i$ and covariance matrix $\Sigma_i$ (here we punt on how to induce the correct number of categories)
The mixture of Gaussians

- Simple generative model of the data: we have $k$ multivariate Gaussians with frequencies $\phi = \langle \phi_1, \ldots, \phi_k \rangle$, each with its own mean $\mu_i$ and covariance matrix $\Sigma_i$ (here we punt on how to induce the correct number of categories)

- $N$ observations are generated i.i.d. by:
  
  \[
  i \sim \text{Multinom}(\phi) \\
  y \sim \mathcal{N}(\mu_i, \Sigma_i)
  \]
The mixture of Gaussians

- Simple generative model of the data: we have $k$ multivariate Gaussians with frequencies $\phi = \langle \phi_1, \ldots, \phi_k \rangle$, each with its own mean $\mu_i$ and covariance matrix $\Sigma_i$ (here we punt on how to induce the correct number of categories)
- $N$ observations are generated i.i.d. by:
  \[
  i \sim \text{Multinom}(\phi) \\
  y \sim \mathcal{N}(\mu_i, \Sigma_i)
  \]
- Here is the corresponding graphical model:
Can we use maximum likelihood?

For observations $\mathbf{y}$ all known to come from the same $k$-dimensional Gaussian, the MLE for the Gaussian’s parameters is

$$
\mu = \langle \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_k \rangle
$$

$$
\Sigma = \begin{bmatrix}
\text{Var}(\mathbf{y}_1) & \text{Cov}(\mathbf{y}_1, \mathbf{y}_2) & \cdots & \text{Cov}(\mathbf{y}_1, \mathbf{y}_k) \\
\text{Cov}(\mathbf{y}_1, \mathbf{y}_2) & \text{Var}(\mathbf{y}_2) & \cdots & \text{Cov}(\mathbf{y}_1, \mathbf{y}_k) \\
\vdots & \vdots & \ddots & \vdots \\
\text{Cov}(\mathbf{y}_1, \mathbf{y}_2) & \text{Cov}(\mathbf{y}_1, \mathbf{y}_2) & \cdots & \text{Var}(\mathbf{y}_k)
\end{bmatrix}
$$

where “Var” and “Cov” are the sample variance and covariance.
Can we use maximum likelihood?

So you might ask: why not use the method of maximum likelihood, searching through all the possible partitions of the data and choosing the partition that gives the highest data likelihood?
Can we use maximum likelihood?

The set of all partitions into \( \langle 3, 3 \rangle \) observations for our example data:
Can we use maximum likelihood?

This looks like a daunting search task, but there is an even bigger problem.
Can we use maximum likelihood?

This looks like a daunting search task, but there is an even bigger problem.
Suppose I try a partition into $\langle 5, 1 \rangle \ldots$
Can we use maximum likelihood?

This looks like a daunting search task, but there is an even bigger problem. 
Suppose I try a partition into $\langle 5, 1 \rangle \ldots$
Can we use maximum likelihood?

This looks like a daunting search task, but there is an even bigger problem.
Suppose I try a partition into $\langle 5, 1 \rangle \ldots$

ML for this partition: $\infty$!!!
Can we use maximum likelihood?

This looks like a daunting search task, but there is an even bigger problem.
Suppose I try a partition into $\langle 5, 1 \rangle$...

More generally, for a $V$-dimensional problem you need at least $V + 1$ points in each partition.
Can we use maximum likelihood?

This looks like a daunting search task, but there is an even bigger problem. Suppose I try a partition into \(\langle 5, 1 \rangle\)...

More generally, for a \(V\)-dimensional problem you need at least \(V + 1\) points in each partition. But this constraint would prevent you from finding intuitive solutions to your problem!
Bayesian Mixture of Gaussians

\[ i \sim \text{Multinom}(\phi) \]
\[ y \sim \mathcal{N}(\mu_i, \Sigma_i) \]
The Bayesian framework allows us to build in explicit assumptions about what constitutes a “sensible” category size.
Bayesian Mixture of Gaussians

\[ i \sim \text{Multinom}(\phi) \]
\[ y \sim \mathcal{N}(\mu_i, \Sigma_i) \]

- The Bayesian framework allows us to build in explicit assumptions about what constitutes a “sensible” category size.
- Returning to our graphical model, we put in a prior on category size/shape.
Bayesian Mixture of Gaussians

\[ i \sim \text{Multinom}(\phi) \]
\[ y \sim \mathcal{N}(\mu_i, \Sigma_i) \]

- The Bayesian framework allows us to build in explicit assumptions about what constitutes a “sensible” category size
- Returning to our graphical model, we put in a prior on category size/shape
Bayesian Mixture of Gaussians

\[ i \sim \text{Multinom}(\phi) \]
\[ y \sim \mathcal{N}(\mu_i, \Sigma_i) \]

- For now we will just leave category prior probabilities uniform:

\[ \phi_1 = \phi_2 = \phi_3 = \phi_4 = \frac{1}{4} \]
Bayesian Mixture of Gaussians

\[ i \sim \text{Multinom}(\phi) \]
\[ y \sim \mathcal{N}(\mu_i, \Sigma_i) \]

- For now we will just leave category prior probabilities uniform:
  \[ \phi_1 = \phi_2 = \phi_3 = \phi_4 = \frac{1}{4} \]

- Here is a conjugate prior distribution for multivariate Gaussians:
  \[ \Sigma_i \sim \mathcal{IW}(\Sigma_0, \nu) \]
  \[ \mu_i | \Sigma \sim \mathcal{N}(\mu_0, \Sigma_i / A) \]
The Inverse Wishart distribution

- Perhaps the best way to understand the Inverse Wishart distribution is to look at samples from it
The Inverse Wishart distribution

- Perhaps the best way to understand the Inverse Wishart distribution is to look at samples from it.
- Below I give samples for \( \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \).
The Inverse Wishart distribution

- Perhaps the best way to understand the Inverse Wishart distribution is to look at samples from it.
- Below I give samples for $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.
The Inverse Wishart distribution

- Perhaps the best way to understand the Inverse Wishart distribution is to look at samples from it.
- Below I give samples for $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- Here, $k = 2$ (top row) or $k = 5$ (bottom row)
Inference for Mixture of Gaussians using Gibbs Sampling

- We still have not given a solution to the search problem
Inference for Mixture of Gaussians using Gibbs Sampling

- We still have not given a solution to the search problem
- One broadly applicable solution is Gibbs sampling
Inference for Mixture of Gaussians using Gibbs Sampling

- We still have not given a solution to the search problem
- One broadly applicable solution is Gibbs sampling
- Simply put:
Inference for Mixture of Gaussians using Gibbs Sampling

- We still have not given a solution to the search problem
- One broadly applicable solution is Gibbs sampling
- Simply put:
  1. Randomly initialize cluster assignments
Inference for Mixture of Gaussians using Gibbs Sampling

- We still have not given a solution to the search problem
- One broadly applicable solution is Gibbs sampling
- Simply put:
  1. Randomly initialize cluster assignments
  2. On each iteration through the data, for each point:
Inference for Mixture of Gaussians using Gibbs Sampling

- We still have not given a solution to the search problem
- One broadly applicable solution is Gibbs sampling
- Simply put:
  1. Randomly initialize cluster assignments
  2. On each iteration through the data, for each point:
     2.1 “Forget” the cluster assignment of the current point $x_i$
Inference for Mixture of Gaussians using Gibbs Sampling

- We still have not given a solution to the search problem
- One broadly applicable solution is Gibbs sampling
- Simply put:
  1. Randomly initialize cluster assignments
  2. On each iteration through the data, for each point:
     2.1 “Forget” the cluster assignment of the current point $x_i$
     2.2 Compute the probability distribution over $x_i$’s cluster assignment conditional on the rest of the partition:

$$P(C_i|x_i, \Pi_{-i}) = \frac{\int_\theta P(x_i|C_i, \theta)P(C_i|\theta)P(\theta) \, d\theta}{\sum_j \int_\theta P(x_j|C_j, \theta)P(C_j|\theta)P(\theta) \, d\theta}$$
Inference for Mixture of Gaussians using Gibbs Sampling

- We still have not given a solution to the search problem
- One broadly applicable solution is Gibbs sampling
- Simply put:

1. Randomly initialize cluster assignments
2. On each iteration through the data, for each point:
   2.1 "Forget" the cluster assignment of the current point $x_i$
   2.2 Compute the probability distribution over $x_i$'s cluster assignment conditional on the rest of the partition:

   $$P(C_i|x_i, \Pi_{-i}) = \frac{\int_{\theta} P(x_i|C_i, \theta)P(C_i|\theta)P(\theta) d\theta}{\sum_j \int_{\theta} P(x_j|C_j, \theta)P(C_j|\theta)P(\theta) d\theta}$$

2.3 Randomly sample a cluster assignment for $x_i$ from $P(C_i|x_i, \Pi_{-i})$ and continue
Inference for Mixture of Gaussians using Gibbs Sampling

We still have not given a solution to the search problem
One broadly applicable solution is Gibbs sampling
Simply put:

1. Randomly initialize cluster assignments
2. On each iteration through the data, for each point:
   2.1 “Forget” the cluster assignment of the current point \( x_i \)
   2.2 Compute the probability distribution over \( x_i \)’s cluster assignment conditional on the rest of the partition:

\[
P(C_i | x_i, \Pi_{-i}) = \frac{\int_{\theta} P(x_i | C_i, \theta) P(C_i | \theta) P(\theta) \, d\theta}{\sum_j \int_{\theta} P(x_j | C_j, \theta) P(C_j | \theta) P(\theta) \, d\theta}
\]

2.3 Randomly sample a cluster assignment for \( x_i \) from \( P(C_i | x_i, \Pi_{-i}) \) and continue
3. Do this for “many” iterations (e.g., until the unnormalized marginal data likelihood is high)
Inference for Mixture of Gaussians using Gibbs Sampling

Starting point for our problem:
One pass of Gibbs sampling through the data
Results of Gibbs sampling with known category probabilities

Posterior modes of category structures:
F1 versus F2 F1 versus Duration F2 versus Duration
Results of Gibbs sampling with known category probabilities

Confusion table of assignments of observations to categories:

Unsupervised

Supervised
The multinomial extension of the beta distribution is the Dirichlet distribution, characterized by parameters $\alpha_1, \ldots, \alpha_k$, and $\mathcal{D}(\pi_1, \ldots, \pi_k)$:

$$\mathcal{D}(\pi_1, \ldots, \pi_k) \overset{\text{def}}{=} \frac{1}{Z} \pi_1^{\alpha_1-1} \pi_2^{\alpha_2-1} \cdots \pi_k^{\alpha_k-1}$$

where the normalizing constant $Z$ is

$$Z = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)\cdots\Gamma(\alpha_k)}{\Gamma(\alpha_1 + \alpha_2 + \cdots + \alpha_k)}$$
Extending the model to learning category probabilities

- So we set:

\[ \phi \sim D(\Sigma_\phi) \]
Extending the model to learning category probabilities

- So we set:

\[ \phi \sim D(\Sigma_\phi) \]

- Combine this with the rest of the model:

\[ \Sigma_i \sim IW(\Sigma_0, \nu) \]
\[ \mu_i | \Sigma \sim N(\mu_0, \Sigma_i / A) \]
\[ i \sim \text{Multinom}(\phi) \]
\[ y \sim N(\mu_i, \Sigma_i) \]
Extending the model to learning category probabilities

- So we set:

\[
\phi \sim \mathcal{D}(\Sigma\phi)
\]

- Combine this with the rest of the model:

\[
\Sigma_i \sim \mathcal{IW}(\Sigma_0, \nu)
\]

\[
\mu_i | \Sigma \sim \mathcal{N}(\mu_0, \Sigma_i/A)
\]

\[
i \sim \text{Multinom}(\phi)
\]

\[
y \sim \mathcal{N}(\mu_i, \Sigma_i)
\]
Extending the model to learning category probabilities

- So we set:

\[ \phi \sim D(\Sigma_\phi) \]

- Combine this with the rest of the model:

\[ \Sigma_i \sim IW(\Sigma_0, \nu) \]
\[ \mu_i | \Sigma \sim N(\mu_0, \Sigma_i / A) \]
\[ i \sim \text{Multinom}(\phi) \]
\[ y \sim N(\mu_i, \Sigma_i) \]
Having to learn category probabilities too makes the problem harder.
Having to learn category probabilities too makes the problem harder.

We can make the problem even more challenging by skewing the category probabilities:

<table>
<thead>
<tr>
<th>Category</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>0.04</td>
</tr>
<tr>
<td>e</td>
<td>0.05</td>
</tr>
<tr>
<td>i</td>
<td>0.29</td>
</tr>
<tr>
<td>I</td>
<td>0.62</td>
</tr>
</tbody>
</table>
Having to learn category probabilities too makes the problem harder.
Having to learn category probabilities too makes the problem harder

Confusion tables for these cases:

With learning of category frequencies

Without learning of category frequencies
Summary

- We can use the exact same models for unsupervised (latent-variable) learning as for hierarchical/mixed-effects regression!
Summary

- We can use the exact same models for unsupervised (latent-variable) learning as for hierarchical/mixed-effects regression!
- However, category induction presents additional difficulties category learning
Summary

- We can use the exact same models for unsupervised (latent-variable) learning as for hierarchical/mixed-effects regression!
- However, category induction presents additional difficulties category learning
  - Non-convexity of the objective function $\rightarrow$ difficulty of search
We can use the exact same models for unsupervised (latent-variable) learning as for hierarchical/mixed-effects regression!

However, category induction presents additional difficulties category learning
  
  - Non-convexity of the objective function $\rightarrow$ difficulty of search
  - Degeneracy of maximum likelihood
Summary

- We can use the exact same models for unsupervised (latent-variable) learning as for hierarchical/mixed-effects regression!

- However, category induction presents additional difficulties category learning
  - Non-convexity of the objective function $\rightarrow$ difficulty of search
  - Degeneracy of maximum likelihood

- In general you need far more data, and/or additional information sources, to converge on good solutions
Summary

- We can use the exact same models for unsupervised (latent-variable) learning as for hierarchical/mixed-effects regression!
- However, category induction presents additional difficulties
  - category learning
    - Non-convexity of the objective function → difficulty of search
    - Degeneracy of maximum likelihood
- In general you need *far* more data, and/or additional information sources, to converge on good solutions
- Relevant references: tons! Read about MOGs for automated speech recognition in Jurafsky and Martin (2008, Chapter 9). See Vallabha et al. (2007) and Feldman et al. (2009) for earlier application of MOGs to phonetic category learning.


