CS 181 Spring 2023 — Section 8

A review of Graphical Models and HMMs

April 11, 2023

Solutions

After reading these notes, you'll be able to:

1. Describe why we use graphical models.
2. Identify Markov chains and when the Markov assumption is valid in real scenarios.
3. Identify when to use a Hidden Markov Model.
4. Identify the *types* of quantities involved in a graphical model.
5. Choose which of the graphical models we've discussed is most suitable to a given machine learning problem.
6. Write down the ELBO for an arbitrary graphical model (without derivations) and formulate the E-step and M-step.

Contents

1 High-level overview

Graphical models are a broad class of models that are useful for *explicitly* representing and inferring the variables that are involved in the data generating process.

They allow us to go *below the surface* and learn **latent variables** that we **can't directly observe**. If you have a guess about the *structure* of the underlying data generating process, then graphical models allow you to **concretely represent** that structure. Once you've inferred the parameters, they can also be used as **generative models** from which you can sample new data points.

However, this also makes inference more difficult: How are we supposed to infer abstract variables that we can't even observe?

We can no longer do maximum likelihood estimation directly. Instead, we make several *approximations* along the way so that solutions to the approximate problem are "good enough".

Expectation maximization (EM) is an algorithm for solving this approximate problem. Instead of maximizing the likelihood directly, we maximize the **evidence lower bound** (ELBO) using **coordinate ascent**.

So now we have a pretty comprehensive strategy for understanding data drawn at a *single* point in time. But what if we want to learn about a *process* that changes from one moment to the next?

We formally represent these as **stochastic processes**. Here, the **state** of the system is modelled by a random variable. The set of values it can take on (aka its *support*) is called the **state space**.

More specifically, we'll focus on processes where each state depends *only* on the previous state, and not on any states further back. This is called the **Markov property** and such processes are called **Markov chains**. You can think of these as graphical models with an arrow pointing from each timestep to the next.

In this course, we'll focus on Markov chains where the **transition probabilities**, i.e. the probability of transitioning from state i to state j, are the same at every timestep. These are called **time-homogenous**.

However, what if we can't observe the entire state? Put another way: What if we think there's some underlying process, responsible for generating our observations, that we can't directly observe?

Hopefully this sounds familiar! Once again, we can treat the state as a **latent variable** that generates our observations. Making this assumption results in a **Hidden Markov Model**.

As before, this makes inference a bit trickier but allows for a great deal more flexibility in the data generating process.

2 Markov chains

A **stochastic process** is a sequence of random variables $S_1, S_2, \ldots$. (Some authors prefer indexing from 0.) Usually the index t represents *time* and the random variable S_t represents the **state** of the system at that time. These random variables take values in the **state space**, notated $\mathcal{S}$, which might be discrete or continuous, finite or infinite.

(Notation: Later on, when the state isn't directly observed, we'll denote it by Z_t.)

A stochastic process satisfies the **Markov property** if the next state depends *only* on the current state, and not any previous states. That is, given S_t, S_{t+1} is conditionally independent from $S_{<t}$.

$$S_{t+1} \mid S_{1:t} \cong S_{t+1} \mid S_t.$$

If a stochastic process satisfies the Markov property, we call it a **Markov chain** (or *Markov model* accordingly).

Exercise: Which of the following could be appropriately modelled as Markov chains?

1. S_t is the result of the t-th coin flip in a sequence of coin flips.
2. S_t is the price of the Tesla stock at t minutes after it went public.
3. S_t is the average user rating of the t-th movie on Netflix when listed alphabetically.
4. Suppose we're playing Snakes and Ladders. S_t is the board after t steps of the game.
5. S_t is the t-th word on the Wikipedia page about Markov chains.

Come up with an example of a Markov chain. Come up with an example of a stochastic process that isn't a Markov chain.

1. An i.i.d. sequence is trivially a Markov chain.
2. This is not a Markov chain. Knowing the past history might help you predict the future price.
3. This is not a Markov chain. In fact, it doesn't make a lot of sense to model this as a stochastic process.
4. This is a Markov chain. The next state of the board depends only on the current state.
5. This is not a Markov chain. The next word depends on the previous words.

2.1 Discrete Markov chains

Let's suppose there are M distinct states, which we'll label 1 through M. That is, $\mathcal{S} = \{1, 2, \ldots, M\}$. Consider the **transition probability** that state i transitions to state j:

$$\mathbb{P}(\text{state } i \to \text{state } j) = \mathbb{P}(S_{t+1} = j \mid S_t = i).$$

We'll assume these probabilities stay the same across time; Such stochastic processes are called **time-homogenous**. We can pack all of these together into an $M \times M$ matrix $\mathbf{T}$ such that

$$T_{ij} = \mathbb{P}(\text{state } i \to \text{state } j).$$

This matrix has a very nice interpretation! The **rows** indicate the "outgoing probabilities" (i.e. if I start in state i, where will I go next?) and the **columns** indicate the "incoming probabilities" (i.e. what are the different ways to get to state j?).

Notably, it gives us a convenient way to take a distribution over states and "push" it forward in time. Let's say that at time t, we have a distribution over the possible states, which we represent as a *row vector* q_t such that $(q_t)_i = \mathbb{P}(S_t = i)$. (CHECK: What properties must this vector satisfy?) Then right-multiplying by the transition matrix gives us the distribution over states at the next time step! (CHECK: Verify this.)

Furthermore, if we want to see what the distribution looks like at time $t + r$, we can simply do this repeatedly! That is, slightly abusing notation for pedagogical clarity, we have

$$\mathbb{P}(S_{t+r} = \cdot \mid S_t \sim q_t) = q_t \mathbf{T}^r,$$

where the l.h.s. is a row vector of probabilities where the ith element is the probability of landing in state i.

Note that the iterated matrix multiplication can be performed efficiently when T is diagonalizable.

2.2 Continuous Markov chains

What if the state is **continuous**? Then to represent the distribution over S_{t+1}, we need to use a probability *density* function (PDF) instead of a probability *mass* function (PMF). This can be described using a **transition kernel**

$$\mathbb{P}(S_{t+1} \approx s' \mid S_t \approx s) = T(s, s').$$

Note that we've used $\approx$ instead of $=$ above; this is because the event $S_t = s$ has *zero probability* for any s since S_t is continuous. Instead, we should think of $S_t \approx s$ as the probability that S_t is in a small neighbourhood of s.

(This is the extent to which you'll need to understand transition kernels for this course.)

CHALLENGE: In the discrete case above, we saw that one can predict the distribution at a future time by repeatedly right-multiplying by the transition matrix. What's the corresponding operation in the continuous case?

In the two-step case, we have (starting from $t = 1$ for simplicity)

$$\mathbb{P}(S_3 \approx s' \mid S_1 \approx s) = \int_{\mathbb{R}} T(s,r)T(r,s')\,dr.$$

This extends to the general case by integrating over all the intermediate steps.

3 Hidden Markov Models

As alluded to in the opening: What if the sequential data we observe isn't the *true* state, but rather just an *observation* derived from the true state?

In this case, we can generalize to **hidden Markov models** in the same way we generalized to latent variable models in the non-sequential case.

That is, we assume that there is some underlying **state process** Z_t (a Markov chain). Then at each timestep, we view some **observation** $Y_t \mid Z_t$ that depends only on the true state at that timestep.

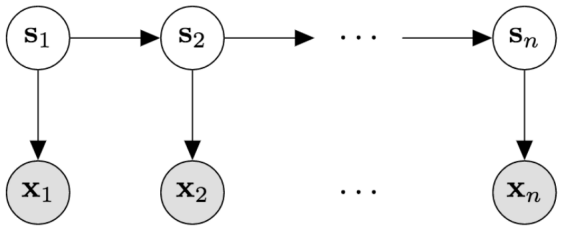

Figure 1: HMM as a graphical model.

HMMs are useful for reasoning about the joint distribution over states and observations:

$$\begin{aligned}\mathbb{P}(z_1, \ldots, z_n, y_1, \ldots, y_n) &= \mathbb{P}(z_1, \ldots, z_n)\mathbb{P}(y_1, \ldots, y_n \mid z_1, \ldots, z_n) \\ &= \mathbb{P}(z_1) \prod_{t=1}^{n-1} \mathbb{P}(z_{t+1} \mid z_t) \prod_{t=1}^{n} \mathbb{P}(y_t \mid z_t)\end{aligned}$$

When the state space is continuous, HMMs are often called **State-Space Models**. Besides numerous applications in physics and control, they've recently also been gaining popularity as language models.

Exercise: Which of these processes would be appropriately represented by a hidden Markov model? For those processes, describe what the latent state process represents.

1. Y_t is the input observed by a sensor on a robot arm at time t.
2. You are on a phone call. Y_t is the audio wave you receive at time t.
3. You are playing rock-paper-scissors and Y_t is the pair of your move and your opponent's move at time t.
4. You are playing poker. Y_t is the set of cards on the board at the t-th turn.

1. The state process could be the position of the robot arm and the observation could be the sensor reading.

2. The state process could be the speaker's intended message and the observation could be the audio wave.

3. There's no need for an HMM in this setting since the state is fully observed (assuming both players act randomly).

4. The state process could be the actual cards in everybody's hand and the observation could be the cards on the board.

3.1 Linear Gaussian Models

As you might have noticed by now, the Gaussian distribution (family) is a popular one to work with because of its several nice properties, including that it's its own conjugate prior and also that a linear combination of independent Gaussians is still Gaussian (even in the multivariate case).

What happens when we make everything Gaussian in the hidden Markov model setting? We get a **linear Gaussian model**. These can be used to describe noisy measurements of a moving object (e.g. missiles, rodents, hands), market fluctuations, etc.

Suppose the state space is $\mathcal{S} = \mathbb{R}^M$ and the observation space is $\mathcal{Y} = \mathbb{R}^O$. Then the state and observation processes are given by

$$\begin{aligned} \text{state process} \quad & \mathbf{Z}_{t+1} \sim \mathbf{A}\mathbf{Z}_t + \mathbf{b} + \mathbf{C}\boldsymbol{\xi}_t, & \boldsymbol{\xi}_t \sim \mathcal{N}(0, \mathbf{I}_M) \\ \text{observation model} \quad & \mathbf{Y}_t \sim \mathbf{D}\mathbf{Z}_t + \mathbf{e} + \mathbf{F}\boldsymbol{\varepsilon}_t, & \boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \mathbf{I}_O). \end{aligned}$$

Exercise: Draw the graphical representation of this HMM. You should have nodes corresponding to $\mathbf{A}, \mathbf{b}, \mathbf{C}, \mathbf{D}, \mathbf{e}, \mathbf{F}, \mathbf{Z}_t, \mathbf{Y}_t, \boldsymbol{\xi}_t, \boldsymbol{\varepsilon}_t$.

Exercise: Infer the shapes of the variables $\mathbf{A}, \mathbf{b}, \mathbf{C}, \mathbf{D}, \mathbf{e}, \mathbf{F}$.

Exercise: Because of the properties of the (multivariate) Gaussian, we know $\mathbf{Z}_{t+1}$ and $\mathbf{Y}_t$ are also (multivariate) Gaussian. Find their means and variances in terms of the above variables (and $\mathbb{Z}_t$).

Challenge: Why might we want to write the system in the form above instead of writing $\boldsymbol{\xi}_t \sim \mathcal{N}(\boldsymbol{\mu}_\xi, \boldsymbol{\Sigma}_\xi)$ for the mean and covariance you described (etc for $\boldsymbol{\varepsilon}_t$)?

We have

$$\begin{aligned} \mathbf{Z}_{t+1} \mid \mathbf{Z_t} &\sim \mathcal{N}(\mathbf{A}\mathbf{Z}_t + \mathbf{b}, \mathbf{C}\mathbf{C}^\top) \\ \mathbf{Y}_t \mid \mathbf{Z_t} &\sim \mathcal{N}(\mathbf{D}\mathbf{Z}_t + \mathbf{e}, \mathbf{F}\mathbf{F}^\top). \end{aligned}$$

We might want to represent the noise parameters explicitly so that we can differentiate w.r.t. the noise parameters.

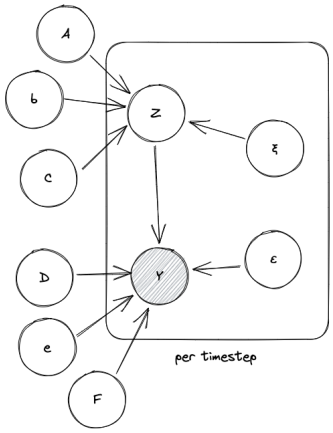

Figure 2: $\mathbf{A}$ is $M \times M$, $\mathbf{b}$ is $M \times 1$, $\mathbf{C}$ is $M \times M$, $\mathbf{D}$ is $O \times M$, $\mathbf{e}$ is $O \times 1$, $\mathbf{F}$ is $O \times O$.

3.2 Solving problems with HMMs

What sorts of tasks might one use an HMM to solve?

(We'll abuse notation and use $y_{1:t}$ to denote the event that we observe $Y_{1:t} = y_{1:t}$.)

1. Learning the *transition probabilities* (aka **dynamics**) of the state or observation models.
2. Inferring the latent variables Z_t from the observations y_t. More specific tasks include:
   (a) **Observation likelihood:** computing $\mathbb{P}(y_{1:n})$.
   (b) **Prediction:** computing $\mathbb{P}(y_{t+1} \mid y_{1:t})$.
   (c) **Filtering:** inferring $Z_t \mid y_{1:t}$ in "real time".
   (d) **Smoothing:** computing Z_t for some *earlier* time than our observed data, i.e. $Z_t \mid y_{1:n}$ where $t < n$.
   (e) Computing the full posterior joint distribution over the sequence of states $Z_{1:n}$ given the observations $y_{1:n}$.

Chapter 10.4 of the CS 181 textbook covers some of these tasks in more detail. Here we give a high-level overview of these algorithms.

3.3 The α and β coefficients (optional)

It turns out that several of the tasks above can be cleanly expressed and solved using two time-indexed quantities denoted α_t, β_t. These are vectors in $[0, 1]^M$.

α_t represents the joint probability of Z_t and all the observations *up to* time t:

$$\alpha_t(j) = \mathbb{P}(Z_t = j, y_{1:t}).$$

β_t represents the likelihood of all the observations *after* time t, conditioned on Z_t:

$$\beta_t(j) = \mathbb{P}(y_{t+1:n} \mid Z_t = j).$$

CHECK: Does β_t define a valid probability distribution over state space? How about α_t?

No. In general the $\beta_t(j)$ do not sum to 1.

Summing the $\alpha_t(j)$ marginalizes out the hidden state, resulting in $\mathbb{P}(y_{1:t})$, which is also not normalized.

These can be visualized as "incorporating information" from the illustrated parts of the graph respectively:

Figure 3: The left image depicts the quantities involved in α_3 and the right image depicts the quantities involved in β_2.

It turns out that there is an efficient dynamic programming implementation for calculating these coefficients. The key idea is that the Markov property allows us to compute α_t and β_t recursively, using the following equations:

$$\alpha_t(j) = \mathbb{P}(y_t \mid Z_t = j) \sum_{k=1}^{M} \mathbb{P}(Z_t = j \mid Z_{t-1} = k) \alpha_{t-1}(k)$$

$$\beta_t(j) = \sum_{k=1}^{M} \mathbb{P}(Z_{t+1} = k \mid Z_t = j) \mathbb{P}(y_{t+1} \mid Z_{t+1} = k) \beta_{t+1}(k).$$

EXERCISE: Derive the two equations above. (This should take around 5 lines of derivation each and make liberal use of the Markov property.)

EXERCISE: Whenever one uses a recursive algorithm, one must specify the base case. What are the base cases for α, β?

EXERCISE: What is $\alpha_t(j)\beta_t(j)$?

EXERCISE: What is the time complexity of the dynamic programming algorithm for computing α_t and β_t?

See the textbook section 10.4.1. The base cases are $\alpha_1(j) = \mathbb{P}(y_1 \mid Z_1 = j)\pi(j)$ (where π is the initial state distribution) and $\beta_n(j) = 1$. (See the textbook for details.)

$\alpha_t(j)\beta_t(j)$ is the joint probability of $Z_t = j$ and all the observations $y_{1:n}$.

The time complexity is $O(M^2 n)$. This is because each $\alpha_t(j)$ and $\beta_t(j)$ requires $O(M)$ time to compute for each of the M possible states at time t, and there are n time steps in total.

Solutions to all of the tasks above (except the last one) can be computed in terms of these coefficients. Try it to get familiar with applying the Markov property!

Observation likelihood	$\mathbb{P}(y_{1:n}) = \sum_{j=1}^{M} \alpha_t(j)\beta_t(j) \quad \text{for any } t$
Prediction	$\mathbb{P}(y_{t+1} \mid y_{1:t}) \propto \sum_{j=1}^{M}\sum_{k=1}^{M} \alpha_t(j)\mathbb{P}(y_{t+1} \mid Z_{t+1} = k)\mathbb{P}(Z_{t+1} = k \mid Z_t = j)$
Filtering	$\mathbb{P}(Z_t = j \mid y_{1:t}) \propto \alpha_t(j)$
Smoothing	$\mathbb{P}(Z_t = j \mid y_{1:n}) \propto \alpha_t(j)\beta_t(j)$

4 Graphical models review

Why are these models called "graphical models"? Here we use "graph" in the *computer science* sense of the term: a set of nodes (representing random variables) connected by directed edges (representing dependency). We draw an arrow from Z to X to mean that X depends on Z.

Representing models as directed graphs allows us to quickly gain information about the variables involved in our model and the dependencies between them.

Here's a chart illustrating the different types of graphical models we've encountered so far:

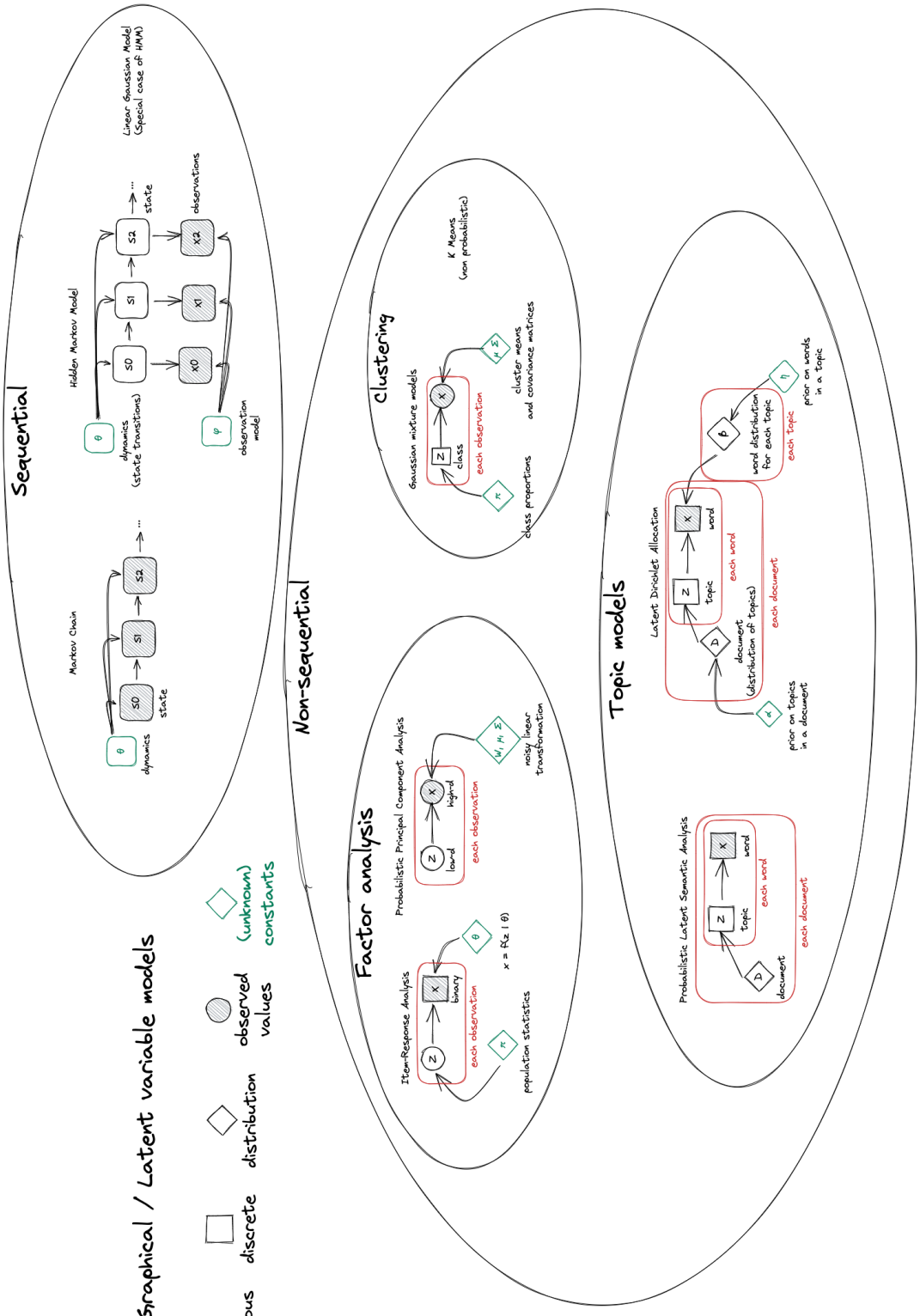

Figure 4: A visual map of several graphical models.

EXERCISE: For each of the models we've studied, identify:

1. Is the observed variable continuous or discrete?
2. If applicable: What likelihood distribution is it drawn from (conditional on the latent variables)?
3. What are the latent variables?
4. If applicable: What prior do we assume they're drawn from?

Name	**Observed variable**	**Likelihood distribution**	**Latent variables**	**Prior distribution**
Probabilistic PCA				
Item-Response Analysis				
Gaussian Mixture Models				
Probabilistic Latent Semantic Analysis				
Latent Dirichlet Allocation				
Markov Models*				
Hidden Markov Models*				
Linear Gaussian Models*				

Table 1: A comparison of various latent variable models.

EXERCISE: Here's a few made-up clients coming to you for aid. For each one: What other information would you need to help them? Which latent variable model(s) would you recommend to answer their question? What are the latent variables? What advantages or disadvantages are there to using the model(s) you suggest? (There isn't necessarily one correct answer to these!)

1. I'm a psychiatrist interested in detecting mental health trends for students in the last few years. So far I've collected some data including their gender, major, alcohol intake, etc, and I ran a linear regression but it didn't fit the data very well. I'm guessing there's some other factors that I didn't include in my survey, but I don't have the budget to run another experiment. Is there any way to identify these factors using my data?
2. I'm a newspaper editor and I have access to a really large database of articles from the last few decades. We recently added a feature that tags each article with a list of

Name	**Observed variable**	**Likelihood distribution**	**Latent variables**	**Prior distribution**
Probabilistic PCA	Continuous	Gaussian	A lower-dimensional representation of the observed data	Gaussian
Item-Response Analysis	Binary	Bernoulli	Flexible	Flexible
Gaussian Mixture Models	Continuous	Gaussian	The class that the given point belongs to	Categorical
Probabilistic Latent Semantic Analysis	Discrete	Categorical	The topic that the word is drawn from; the document that the topic is drawn from	Both categorical
Latent Dirichlet Allocation	Discrete	Categorical	The above; the distribution of topics found in the document; the distribution of words belonging to that topic	Both Dirichlet
Markov Models*	Either	Depends	None	
Hidden Markov Models*	Same	Same	The state underlying the observations	Flexible
Linear Gaussian Models*	Continuous	Gaussian	Same as observed variable	Gaussian

Table 2: A comparison of various latent variable models.

related topics. Right now we have some interns labelling each of our new articles by hand. It would be really great if we could go back and label all of the historical articles too, but there's no way we'd ever get through all of them. Is there a way to do that automatically?

3. After watching Cloudy with a Chance of Meatballs, I got really interested in meteorology and so I've been trying to predict the weather using a Markov chain. I've tried including a lot of variables, including the humidity, wind speeds, and even the ratings at local restaurants, but it just doesn't seem to work. I thought that if I included more variables that my model would get better, but somehow it's actually been getting worse! Can you help me fix it?

4. I'm a financial analyst trying to analyze how recent changes in economic conditions have been affecting different regions of the market. I'm trying to figure out a way to classify stocks that doesn't require me to go through and label them manually. The only information I have is the daily price of each stock over the last 10 years. Is there some way I could classify these stocks based only on their price data?

Come up with at least one other scenario.

Here are some possible solutions:

1. One could use a Gaussian mixture model with a certain number of classes and see if that's enough to increase the fit of the model.
2. One could try Latent Dirichlet Analysis or other topic models.
3. This might be because of the Bias-Variance tradeoff. One could try to reduce the number of variables included in the model and increase the length of time considered in the state.
4. One could try a combination of PCA and clustering algorithms to classify the stocks. There might also be room for a creative use of HMMs to learn a useful representation of the data.

4.1 Identifying ELBO and EM steps from a graphical representation

Let's do a high-level sweep of the EM algorithm. It's *crucial* to keep in mind the *types* of the various quantities involved in the calculation. In particular, there are three relevant *types* of quantities:

1. The **observed variables** X,
2. The unobserved **latent variables** Z, and
3. The unknown **constants** θ that parameterize the data generating process.

Here, the symbols X, Z, and θ are not meant to refer to specific quantities, but rather as representatives of the *types* described above. For instance, our model might involve many different parameters; then we should simply substitute them in for θ.

These quantities can easily be read off of a graphical model.

In terms of these abstract types, the ELBO can be expressed as

$$\text{ELBO} = \mathop{\mathbb{E}}_{Z\sim q} \log \frac{\mathbb{P}(X, Z \mid \theta)}{q(Z)}.$$

Exercise: Write the ELBO terms for Hidden Markov Models and Latent Dirichlet Allocation.

For Hidden Markov Models, one has

$$\text{ELBO} = \mathop{\mathbb{E}}_{z_{1:n}\sim q} \log \frac{\mathbb{P}(z_{1:n}, y_{1:n} \mid \theta)}{q(z_{1:n})}$$

where θ denotes some set of parameters responsible for the dynamics or for observation noise.

For LDA, using our earlier notation, one has the observed variables W, the latent variables D, Z, β, and the constants α and η. Suppose there are n documents, m words per document, and k topics. The ELBO is then

$$\text{ELBO} = \mathop{\mathbb{E}}_{d_{1:n}, z_{1:nm}, \beta_{1:k} \sim q} \log \frac{\mathbb{P}(w_{1:nm}, d_{1:n}, z_{1:nm}, \beta_{1:k} \mid \alpha, \eta)}{q(d_{1:n}, z_{1:nm}, \beta_{1:k})}.$$

Assuming independence, one could also expand out the log-likelihood into a sum over the individual terms.

The EM algorithm simply optimizes the unknown quantities Z and θ in turn. The observed variables X are fixed throughout the whole optimization process.

In the **E-step**, given the *parameters*, we infer a *distribution* over the *latent variables*:

$$E : \theta^t \mapsto q^t(Z).$$

Specifically, we set $q^t(Z)$ to be the *posterior* distribution $\mathbb{P}(Z \mid X, \theta^t)$. (We showed in class that this is optimal for maximizing the ELBO.)

In the **M-step**, we optimize the ("constant") *parameters* given the *latent variables*:

$$M : q^t(Z) \mapsto \theta^{t+1}.$$

Then since we hold q^t fixed, we can ignore it in the ELBO:

$$\theta^{t+1} = \arg\max_{\theta} \mathop{\mathbb{E}}_{Z \sim q^t} \log \mathbb{P}(X, Z \mid \theta).$$

And that's the core of EM.

Though we'd prefer models for which both of these steps can be solved analytically, often we also use complex models for which this is not possible. In these cases, we can use numerical methods to solve the E-step and M-step. Specifically, we can use **variational methods** to approximate the posterior during the E-step, and **gradient descent** to optimize the parameters during the M-step.

Exercise: Write the E-step and M-step (no need to expand further) for Hidden Markov Models and Latent Dirichlet Allocation.

For Hidden Markov Models, the E-step is

$$q^t(z_{1:n}) = \mathbb{P}(z_{1:n} \mid y_{1:n}, \theta^t).$$

The M-step is

$$\theta^{t+1} = \arg\max_{\theta} \mathop{\mathbb{E}}_{z_{1:n} \sim q^t} \log \mathbb{P}(z_{1:n}, y_{1:n} \mid \theta).$$

For LDA, the E-step is

$$q^t(d_{1:n}, z_{1:nm}, \beta_{1:k}) = \mathbb{P}(d_{1:n}, z_{1:nm}, \beta_{1:k} \mid w_{1:nm}, \alpha^t, \eta^t).$$

The M-step is

$$\alpha^{t+1}, \eta^{t+1} = \arg\max_{\alpha,\eta} \mathop{\mathbb{E}}_{d_{1:n}, z_{1:nm}, \beta_{1:k} \sim q^t} \log \mathbb{P}(w_{1:nm}, d_{1:n}, z_{1:nm}, \beta_{1:k} \mid \alpha, \eta).$$