STA414/2104: Week 7

Discrete Latent Variable Models

Alex Stringer

Feb 27, 2018

Recap

So far we have talked about models for continuous and discrete data.

Our approach was

- Specify a probability distribution for our data
- Decompose the parameters of the distribution into a function of features and (other) parameters
- Find estimates of the parameters using maximum likelihood, by maximizing a posterior, or some other method
- Use the results to make predictions for new points

Example

For example, (binary) logistic regression. We have

$$t_n \sim Binom(1, p_n)$$

and we observe a training set of *independent* observations $(t_n, \mathbf{x}_n)$.
We write

$$p_n = \frac{1}{1 + \exp(-\mathbf{x}_n' \mathbf{w})}$$

or

$$\mathbf{x}_n' \mathbf{w} = \log \left(\frac{p_n}{1 - p_n} \right)$$

Recap

Under that approach, we specify all of the structure in the training data through the probability distribution we pick.

We specify everything about the functional relationship between features and targets directly. Usually we use a linear combination somewhere.

What if we think our data has additional structure?

Additional Structure

The strongest assumption we make about the sample is that it's *independently, identically distributed* (given the features).

We also assume that the p features in the dataset all contribute equally to the distribution of $\mathbf{x}$.

We can relax these in several ways:

- Not independent; observations are correlated somehow
- The distribution of $\mathbf{x}$ lies close to some $d < p$-dimensional subspace of the p-dimensional feature space
- Not identically distributed; observations come from one of $K > 1$ *component distributions*

We will talk about the third option today, and use this to motivate the concept of a *latent variable*.

Grouped Data

Suppose we observe only features $\mathbf{x}_n, n = 1 \dots N$, no targets.

We postulate that each feature vector was generated from one of $K > 1$ component densities, or *groups*.

We want to develop a procedure for *classifying* each training vector into one of the groups.

This is called *unsupervised classification*, or *clustering*.

Grouped Data

This is an example of a situation in which we believe that there is a certain structure in our dataset that goes beyond the usual IID case.

Let's first look at how we might approach this problem directly, with a special-purpose algorithm.

K-Means

In the *supervised* classification case, we had K-Nearest-Neighbours:

- For each training set point, select the K closest points in the training set
- Average their targets (could be either continuous or discrete)

We can develop an analagous procedure when we don't know the class labels of the points in the training set

K-Means

The K-Means algorithm is as follows:

- Choose K
- Initialize mean vectors $\boldsymbol{\mu}_k, k = 1 \ldots K$
- Define variables $z_{nk} \in \{0, 1\}$ which code whether observation n belongs to class k

We then define an objective function to minimize:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

K-Means

We fit the algorithm by iterating between the following steps:

- Given the $\boldsymbol{\mu}_k$, choose z_{nk} to minimze J. This is obtained by

$$z_{nk} = \begin{cases} 1 \text{ if } k = argmin_j \left\| \mathbf{x}_n - \boldsymbol{\mu}_j \right\|^2 \\ 0 \text{ else} \end{cases}$$

- Given the z_{nk}, set

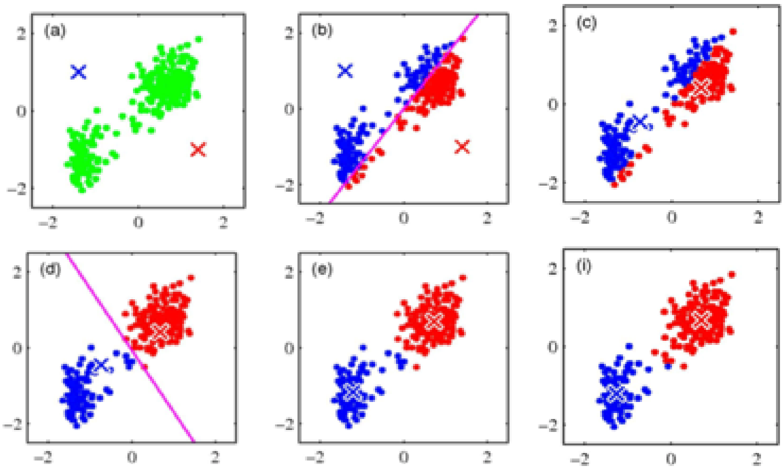
$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N z_{nk}\mathbf{x}_n}{\sum_{n=1}^N z_{nk}}$$

  the sample mean of the k^{th} group.

Note that $\sum_{n=1}^N z_{nk}$ is just the number of points currently assigned to group K.

Illustration

Illustration of using 2-Means on the Old Faithful dataset (Bishop (2006)):



Properties

The K-Means algorithm was an early solution to the unsupervised classification problem. It has the advantage of converging very fast, and of providing a clear, interpretable modelling procedure.

Properties

It's not perfect:

- You have to choose good starting values, which in practice means run the algorithm many times with different randomly generated starting values (which are themselves vectors)
- You have to choose K, the number of groups. In practice people usually repeat the procedure for many values of K, then pick the "best" one (what is "best"?)
- It provides only "hard" classifications. If you think that the component densities are well separated, this might be fine, but especially in higher-dimensional situations, there may be more than one plausible component for each $\mathbf{x}_n$ to belong to

Latent Variables

It is not ideal that every time we think of some structure our data might exhibit, we have to come up with a special purpose-built algorithm for making predictions.

We might want to modify our approach to be more flexible.

To do this, we introduce the concept of a **latent variable**.

Latent means *unobserved*.

Latent Variables

Latent variables are entites that we invent to explain patterns we see in observable variables- for instance, doctors have invented diseases to explain commonalities of symptoms in patients

`- Radford Neal`

We infer the values of these latent variables from the observed data.

Mixture Models

In our clustering example, let's define z_{nk} as before, but this time consider these to be *unobserved random variables* rather than fixed constants.

The z_{nk} are **discrete latent variables**.

We can model the z_{nk} directly by writing

$$\mathbf{z}_n = (z_{n1}, \ldots, z_{nK})$$

and specifying $\mathbf{z}_n$ to have a multinomial distribution with parameters $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$,

$$p(\mathbf{z}_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$$

Mixture Models

We have just put structure into our problem: we have formally stated our belief that, prior to actually observing $\mathbf{x}_n$, we think it belongs to group k with probability π_k, $k = 1 \dots K$.

The goal of our analysis is to predict $\mathbf{z}_n$ having observed $\mathbf{x}_n$.

We can specify the conditional distribution of $\mathbf{x}_n$ given $\mathbf{z}_n$:

$$p(\mathbf{x}_n|\mathbf{z}_n) = \prod_{k=1}^{K} p_k(\mathbf{x}_n|\theta_k)^{z_{nk}}$$

Mixture Models

Because z_{nk} equals 1 when $\mathbf{x}_n$ comes from group k and zero else, this is just a way of writing

$$p(\mathbf{x}_n|\mathbf{z}_n) = p_k(\mathbf{x}_n|\theta_k) \text{ if } \mathbf{x}_n \in \text{group k}$$

In principle the component densities $p_k(\mathbf{x}_n|\theta_k)$ could be anything, and don't have to all be from the same family.

In this course, as is common, we will assume each component density is Gaussian:

$$p_k(\mathbf{x}_n|\theta_k) = N(\boldsymbol{\mu}_k, \Sigma_k)$$

Mixture Models

We have therefore specified a full joint distribution of observed and latent variables

$$\begin{aligned} p(\mathbf{x}_n, \mathbf{z}_n) &= p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n) \\ &= \prod_{k=1}^{K} \left(\pi_k N(\boldsymbol{\mu}_k, \Sigma_k)\right)^{z_{nk}} \end{aligned}$$

However, we don't actually observe $\mathbf{z}_n$; they are latent. What we need is to find

$$p(\mathbf{z}_n|\mathbf{x}_n) = \frac{p(\mathbf{x}_n, \mathbf{z}_n)}{p(\mathbf{x}_n)}$$

the *posterior probability* that $\mathbf{x}_n$ belongs to each group $k = 1 \dots K$.

Relationship to Bayesian

We're using the same formulas and words to describe these quantities as we did when we studied Bayesian inference.

There are some conceptual differences. Here, we literally believe that there is some random quantity z_{nk} that represents the group membership of $\mathbf{x}_n$. And we don't really have much choice in specifying its distribution; under our model, the multinomial distribution captures our beliefs.

In the Bayesian framework, we were using probability distributions directly to represent uncertainty; we didn't necessarily think that the parameters on which we were putting prior distributions were actually random variables.

Mixture Models

We can compute the marginal distribution of $\mathbf{x}_n$ now (exercise: verify this):

$$p(\mathbf{x}_n) = \sum_{\mathbf{z}} p(\mathbf{x}_n, \mathbf{z}_n) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{\mu}_k, \Sigma_k)$$

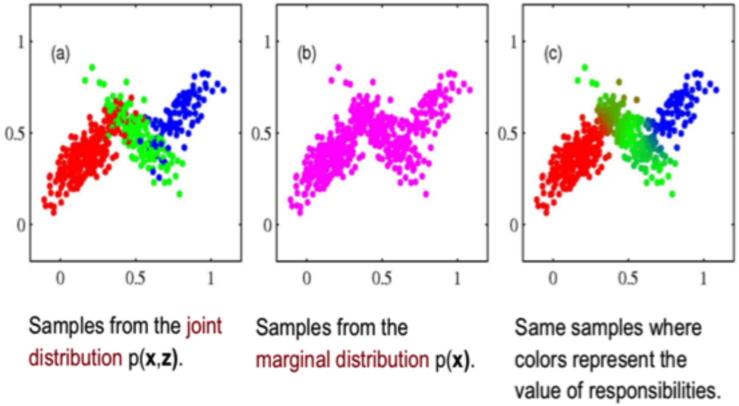
which we interpret as a *mixture* of the component densities, leading to the term **mixture model**.

Note that

- The sum is taken over all of the K possible configurations of $\mathbf{z}_n$, each of which assigns 1 to exactly one group and 0 to the others
- We introduced $\boldsymbol{\pi}$ as being multinomial parameters, which means $\sum_{k=1}^{K} \pi_k = 1$. This confirms that the above expression actually is a density

Example

Example of a mixture of 3 Gaussians (from Bishop (2006)):



Mixture Models

With all this in hand, we compute the posterior probability of $\mathbf{x}_n$ belonging to each group, which gives us *soft* classifications. Note that Bishop refers to these as "responsibilities", as in the previous plot.

We take the highest posterior probability as the *hard* $(0/1)$ classification for each point.

After introducing this stucture, we have again reduced the modelling problem to one of parameter estimation: we need to estimate the $\boldsymbol{\mu}_k, \Sigma_k, k = 1 \ldots K$.

Let's try maximum likelihood.

Maximum Likelihood

The likelihood takes the form (we're still assuming independence between the $\mathbf{x}_n$)

$$L(\boldsymbol{\theta}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k N(\boldsymbol{\mu}_k, \Sigma_k)$$

where the full parameter vector $\boldsymbol{\theta}$ consists of the $\boldsymbol{\mu}_k$ and the unique elements of the Σ_k.

The log-likelihood is therefore

$$\ell(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k N(\boldsymbol{\mu}_k, \Sigma_k) \right)$$

Maximum Likelihood

If you try and differentiate this with respect to each $\pi_k, \boldsymbol{\mu}_k, \Sigma_k$, set equal to zero, and solve, you will obtain

$$
\begin{aligned}
\hat{\pi}_k &= \frac{\sum_{n=1}^N \hat{z}_{nk}}{N} \\
\hat{\boldsymbol{\mu}}_k &= \frac{\sum_{n=1}^N \hat{z}_{nk}\mathbf{x}_n}{\sum_{n=1}^N \hat{z}_{nk}} \\
\hat{\Sigma}_k &= \frac{1}{\sum_{n=1}^N \hat{z}_{nk}} \sum_{n=1}^N \hat{z}_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)'
\end{aligned}
$$

where

$$
\hat{z}_{nk} = \frac{\hat{\pi}_k N(\hat{\boldsymbol{\mu}}_k, \widehat{\Sigma_k})}{\sum_{j=1}^K \hat{\pi}_j N(\hat{\boldsymbol{\mu}}_j, \hat{\Sigma}_j)} = p(z_{nk}|\mathbf{x}_n)
$$

Maximum Likelihood

These solutions look remarkably similar to the formulas obtained by K-means.

However like with K-means, there is no closed-form answer here, because the $\hat{z}_{nk}$ depend on the values of the parameters.

Connection to K-means

It is not a coincidence that the formulas look similar to K-means.

Set $\Sigma_k = \sigma^2 \mathbf{I}$ for all k, i.e. assume the groups have equal, spherical covariance with radius σ^2.

We have

$$\hat{z}_{nk} = \frac{\pi_k \exp(-\left\|\mathbf{x}_n - \boldsymbol{\mu}_k\right\|^2 / 2\sigma^2)}{\sum_{j=1}^{K} \pi_j \exp(-\left\|\mathbf{x}_n - \boldsymbol{\mu}_j\right\|^2 / 2\sigma^2)}$$

Now let $\sigma^2 \to 0$. The term for which $\left\|\mathbf{x}_n - \boldsymbol{\mu}_k\right\|^2$ is smallest will go to zero most slowly, so

$$\hat{z}_{nk} \to \begin{cases} 1 \text{ if } k = argmin \left\|\mathbf{x}_n - \boldsymbol{\mu}_k\right\|^2 \\ 0 \text{ else} \end{cases}$$

Maximum Likelihood

We could maximize the likelihood using gradient descent. It is, though, a pretty complicated likelihood.

In particular, it has many local maxima, and is unbounded if one of the clusters contains only a single point (why?).

You can play around with it using Autograd:

https://github.com/HIPS/autograd/blob/master/examples/gmm.py

EM Algorithm

There is a better way. We said that latent variables are random variables that we don't observe. We can regard them, then, as **missing data**.

The **EM Algorithm** is an algorithm for finding maximum likelihood estimates for parameters in the presence of missing data.

This was published in 1977 with the original intent being to account for actual missing data in experiments.

It has gained tremendous popularity in our current situation: we deliberately introduced missing/latent data to induce structure into our model

EM Algorithm

Suppose we have a vector $\mathbf{y}$ which we partition into components that we *observe*, $\mathbf{x}$, and components that are *missing*, $\mathbf{z}$, so $\mathbf{y} = (\mathbf{x}, \mathbf{z})$. We use the following terminology:

- $\mathbf{y}$: **complete data**
- $\mathbf{x}$: **observed data**
- $\mathbf{z}$: **missing data**

In mixture models, $\mathbf{x}$ are the features/observations and $\mathbf{z}$ are the group memberships, as we have been saying so far.

We call the joint likelihood of $(\mathbf{x}, \mathbf{z})$ the *complete-data likelihood*:

$$L(\theta|\mathbf{x}, \mathbf{z})$$

EM Algorithm

The EM Algorithm proceeds as follows. At iteration t:

- **E-Step**: compute the *expected* complete-data log-likelihood with respect to the missing data, conditional on the observed data, using the current parameter estimates:

$$Q(\theta, \theta^t) = E_{\mathbf{z}|\mathbf{x},\theta^t}\left(\log L(\theta|\mathbf{x}, \mathbf{z})\right)$$

- **M-Step**: maximize this with respect to θ to obtain θ^{t+1}:

$$\theta^{t+1} = \text{argmax}_\theta Q(\theta, \theta^t)$$

Expected Complete Data Log Likelihood $Q(\theta, \theta^t)$

In words, we refer to this function as the "expected complete-data log-likelihood conditional on the observed data and current parameter estimates".

- The expectation, $E_{\mathbf{z}|\mathbf{x},\theta^t}$, is taken with respect to the missing data $\mathbf{z}$, conditional on the complete data $\mathbf{x}$, at the current parameter estimates θ^t
- The log-likelihood inside the expectation is a function of $\mathbf{z}$, $\mathbf{x}$, and θ
- When we take its expectation, the result is no longer a function of $\mathbf{z}$
- The result, though, will now be a function of *both* θ and θ^t (as well as $\mathbf{x}$), because we took the expectation conditional on $\mathbf{x}$ and θ^t.

Maybe an example would help?

Example: Multinomial Cell Probabilities

Here is one of the motivating examples presented in the original paper on EM (Dempster, Laird and Rubin, 1977: Maximum Likelihood from Incomplete Data via the EM Algorithm).

Suppose we observe $n = 197$ animals grouped into four groups of some kind, with probabilities

$$\left(\frac{1}{2} + \frac{\theta}{4}, \frac{1-\theta}{4}, \frac{1-\theta}{4}, \frac{\theta}{4}\right)$$

and we observe

$$\mathbf{x} = (x_1, x_2, x_3, x_4) = (125, 18, 20, 34)$$

Example: Multinomial Cell Probabilities

This is not a missing data problem (yet). The complete log likelihood is

$$\begin{aligned}\ell(\theta) = const + x_1 \log\left(\frac{1}{2} + \frac{\theta}{4}\right) + x_2 \log\left(\frac{1-\theta}{4}\right) \\ + x_3 \log\left(\frac{1-\theta}{4}\right) + x_4 \log\left(\frac{\theta}{4}\right)\end{aligned}$$

which you can differentiate and solve (it's a messy quadratic) to obtain the MLE $\hat{\theta}$

Example: Multinomial Cell Probabilities

But now, suppose the scientists come back and tell you: group 1 is actually *two* groups, which occur with probabilities $1/2$ and $\theta/4$ respectively.

The complete data is then

$$y = (\mathbf{x}, \mathbf{z}) = (z_{11}, z_{12}, x_2, x_3, x_4)$$

where $z_{11} + z_{12} = x_1$ was observed.

Example: Multinomial Cell Probabilities

The complete-data log-likelihood is then

$$\begin{aligned}\ell(\mathbf{x}, \mathbf{z}|\theta) = const + z_{12} \log\left(\frac{\theta}{4}\right) + x_2 \log\left(\frac{1-\theta}{4}\right) \\ + x_3 \log\left(\frac{1-\theta}{4}\right) + x_4 \log\left(\frac{\theta}{4}\right)\end{aligned}$$

where we ignore (in the constant) all terms not depending on θ.

We can't maximize this directly, since we don't observe z_{12}.

Example: Multinomial Cell Probabilities

Initialize $\theta = \theta^0$.

Compute the expected complete-data log-likelihood conditional on the observed data and θ^0:

$$
\begin{aligned}
Q(\theta, \theta^0) = E_{\mathbf{z}|\mathbf{x},\theta^0} &\left(z_{12} \log\left(\frac{\theta}{4}\right) + x_2 \log\left(\frac{1-\theta}{4}\right) + x_3 \log\left(\frac{1-\theta}{4}\right)\right. \\
&\left. + x_4 \log\left(\frac{\theta}{4}\right) | x_1, x_2, x_3, x_4, \theta^0 \right) \\
&= E(z_{12}|x_1, \theta^0) \log\left(\frac{\theta}{4}\right) + x_2 \log\left(\frac{1-\theta}{4}\right) \\
&+ x_3 \log\left(\frac{1-\theta}{4}\right) + x_4 \log\left(\frac{\theta}{4}\right)
\end{aligned}
$$

Example: Multinomial Cell Probabilities

Because conditional on $x_1, \ldots, x_4$ and θ^0, z_{12} is the only term that is random, our problem is reduced to computing

$$E(z_{12}|x_1, \theta^0)$$

That is, given our current estimate of the parameter θ, and the fact that we saw x_1 animals across cells z_{11} and z_{12}, how many of those animals do we think were in cell z_{12}?

Example: Multinomial Cell Probabilities

Conditional on $x_1 = z_{11} + z_{12}$, z_{12} follows a binomial distribution,

$$z_{12}|x_1, \theta^0 \sim Binom\left(x_1, \frac{\theta^0/4}{1/2 + \theta^0/4}\right)$$

Which means

$$E(z_{12}|x_1, \theta^0) = \frac{\theta^0 x_1}{2 + \theta^0} \equiv \hat{z}_{12}$$

Example: Multinomial Cell Probabilities

The expected complete-data log-likelihood is then

$$Q(\theta, \theta^t) = \hat{z}_{12} \log\left(\frac{\theta}{4}\right) + x_2 \log\left(\frac{1-\theta}{4}\right) + x_3 \log\left(\frac{1-\theta}{4}\right) + x_4 \log\left(\frac{\theta}{4}\right)$$

Note that θ appears through the terms originally in the log-likelihood, and θ^0 appears silently through $\hat{z}_{12}$.

Maximizing with respect to θ gives

$$\theta^{t+1} = \frac{\hat{z}_{12}^t + x_4}{\hat{z}_{12}^t + x_2 + x_3 + x_4}$$

The algorithm would then iterate between calculating $\hat{z}_{12}$ and θ until convergence to a local maximum.

EM Algorithm

In general, there is no guarantee that we would be able to perform these calculations easily, or at all.

However, it works out nicely in one special case: when the complete-data log-likelihood is *linear* in the missing data.

This happens remarkably often, including in the above example, and in mixture models.

EM Algorithm

For mixture models, the complete-data likelihood is

$$p(\mathbf{X}|\mathbf{Z}) = \prod_{n=1}^{N} p(\mathbf{x}_n, \mathbf{z}_n) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\pi_k N(\boldsymbol{\mu}_k, \Sigma_k)\right)^{z_{nk}}$$

The complete-data log-likelihood is then

$$\log p(\mathbf{X}|\mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log \left(\pi_k N(\boldsymbol{\mu}_k, \Sigma_k)\right)$$

which is a linear function of the missing data $\mathbf{Z}$.

EM Algorithm

This lets us compute

$$
\begin{aligned}
Q(\theta, \theta^t) &= E_{\mathbf{z}|\mathbf{x}}\Big(\log L(\theta^t|\mathbf{x}, \mathbf{z})\Big) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} E_{\mathbf{z}|\mathbf{x}}(z_{nk}) \log\left(\pi_k N(\boldsymbol{\mu}_k, \Sigma_k)\right)
\end{aligned}
$$

so the E-Step amounts to replacing the z_{nk} with their current conditional expectation given the observed data and parameter estimates,

$$E_{\mathbf{z}|\mathbf{x}}(z_{nk}) = \hat{z}_{nk}$$

EM Algorithm

The M-Step is then to maximize this quantity- but we already did that, so really the M-Step is to compute the MLEs we already derived, but plugging in $\hat{z}_{nk}$ for z_{nk}:

$$\hat{\pi}_k = \frac{\sum_{n=1}^{N} \hat{z}_{nk}}{N}$$

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{n=1}^{N} \hat{z}_{nk}\mathbf{x}_n}{\sum_{n=1}^{N} \hat{z}_{nk}}$$

$$\hat{\Sigma}_k = \frac{1}{\sum_{n=1}^{N} \hat{z}_{nk}} \sum_{n=1}^{N} \hat{z}_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)'$$

Note this gives us a motivation for the K-Means procedure we stated previously.

Predictions

Once the EM Algorithm converges, we can use the final parameter estimates to predict the class of each point.

This gives us soft classifications, which we make hard (0/1) by taking the class with the highest predicted probability for each point.