CPSC 540 Assignment 5 (due April 10)

UGMs, Bayes, and Literature Survey

1 Undirected Graphical Models

1.1 Conditional UGM

Consdier modeling the dependencies between sets of binary variables x_j and y_j with the following UGM which is a variation on a stacked RBM: Computing univariate marginals in this model will be NP-hard in general, but the graph structure allows efficient block updates by conditioning on suitable subsets of the variables (this could be useful for designing approximate inference methods). For each of the conditioning scenarios below, draw the conditional UGM and comment on how expensive it would be to compute univariate marginals in the conditional UGM.

1. Conditioning on all the z and h values.
2. Conditioning on all the x and h values.
3. Conditioning on all the z and y values.
4. Conditioning on all the x and z values.

Solutions

1. By conditioning on z and h we get the following UGM

You have to show the resulting UGM

In this case since we have a fully disconected graph, estimating the univariate marginals is trivial.

2.
In this case the UGM that we obtain is shown below

Present the UGM when conditioning on x and h

As before we have a fully disconnected graph, i.e. all variables are independent, hence estimating the marginals is trivial too.

3.
In this case we have the following UGM when we condition on z and y we get the following UGM

Present the UGM when conditioning on z and y

In this case we have that the x_i form a fully disconected graph, hence calculating the univariate marginals in the x variables is straightforward. For the h variables the situation is more interesting since we have a chain structure. In this case the three width is $\omega = 1$, hence the cost of calculating the marginals is $\mathcal{O}(dk^{\omega+1}) = \mathcal{O}(4 * 2^2)$.

4.

1.2 Fitting a UGM to PINs

The function *example_UGM* loads a dataset X containing samples of PIN numbers, based on the probabilities from the article at this URL: `http://www.datagenetics.com/blog/september32012`.[1]

This function shows how to use the UGM software to fit a UGM model to the dataset, where all node/edge parameters are tied and the graph is empty. It then performs decoding/inference/sampling in the fitted model. Unfortunately, this is not a very good model of the data for several reasons:

1. The decoding is 1 1 1 1, whereas in the data the most likely value by far is 1 2 3 4. Similarly, the sampler doesn't tend to generate 1 2 3 4 even though this happens in more than 1/10 samples.
2. The marginal probability of the first number being 1 is 22.06%, which is acutally too low (it should be 38.54%). In addition, the marginal probabilities of the remaining nubmers being 1 are also 22.06%, and these numbers are too high.
3. Conditioned on the first three numbers being 1 2 3, the probability that the last number is 4 is less than 10% in the model, whereas in the data it's more than 90%.

In this question you'll explore better models of this data and different aspects of UGMs.

1. Why does w have a length of 9?
2. Write an equation for the model being used by the code.
3. What are potential sources of the problems above?
4. Modify the demo to use a *tied* value of 0 and re-run the demo. Why does the model now have 36 parameters? Comment on whether this fixes each of the above 3 issues.
5. Modify the demo to use chain-structured dependency (keeping the *tied* value at 0). Comment on whether this fixes each of the above 3 issues.
6. Modify the demo to use a completely-connected graph (keeping the *tied* value at 0). Comment on whether this fixes each of the above 3 issues.
7. UGM only support pairwise graphs, what would the effect of higher-order potentials be? What would the disdavantages of higher-order potentials be?

If you want to further explore UGMs, there are quite a few demos on the UGM webpage that you can go through which cover all sorts of things like approximate inference and CRFs.

Solution

1.
Our state space has $k = 10$ elements (numbers from 0 to 9). We can always fix one of the values $w_j = 0$ (why??). Hence in our case, we need to find only $10 - 1 = 9$ values of the vector w.

2.

[1] I got the probabilities from the reverse-engineered heatmap here: `http://jemore.free.fr/wordpress/?p=73`.

Since we are using a log-linear model for $x = [x^1, x^2, x^3, x^4]^T$, with independence among the variables i.e. we have that the set of edges in the UGM is empty. Then the model we are using is of the form

$$\mathbb{P}(x|w) = \prod_{i=1}^{4} \mathbb{P}(x^i|w) = \frac{1}{Z(w)} \exp(-\sum_{i=1}^{4} w^T F(x^i)).$$

In our case we have $w = [w_1, \dots, w_9]^T$ and $F(x^i) = [F_1(x^i), \dots F_9(x^i)]$, where

$$F_j(x^i) = I(x^i = j) \qquad \text{for } j = 1, 2, \dots, 9 \text{ and } i = 1, 2, 3, 4.$$

3.

Since we are using a fully disconnected graph as our model, this means we assume independence between the variables. This means that the model can be written in term of potentials as

$$p(x|w) \propto \prod_{j=1}^{4} \phi_j(x^j),$$

with $\phi_j(k) = \exp(w_k)$ for $k = 1, 2, \dots, 9$ and all j. Hence, when doing decoding we have

$$\max_x \mathbb{P}(x|w) = \max_{x^1} \phi_1(x^1) \max_{x^2} \phi_2(x^2) \max_{x^3} \phi_3(x^3) \max_{x^4} \phi_4(x^4).$$

and from running the script example_UGM.m we know that the maximum for each of the potential functions occurs when $\phi_j(1) = 1.6722$ for all j. Hence by doing a decoding under this model we would conclude that the most likely values is $1, 1, 1, 1$. Which is clearly wrong. This result also points to a second and a third weakness in the model, namely, we are missing all correlations between digits. For instance if the first 3 digits are $1, 2, 3$ we know from the data that the most likely number to continue the sequence would be 4. But since the model assumes all variables to be independent, we missed that. Also when calculating the marginals we ommit all contributions from the other digits, for example for x_1 we have

$$p(x_1|w) = \frac{\phi_1(x^1)}{Z(w)} \prod_{j=2}^{4} \sum_{x^j} \phi_j(x^j).$$

So here we also missed the connections that relate the different digits.

4.

When setting the parameter *tied* $= 0$ now each potential function satisfy

$$\log(\phi_i(x^j) = w_{ij}$$

Again we have the freedom to fix one of the values one of the 10 spaces. In this case the potential function for the value $k = 0$ is fixed at 1 hence for the other 9 potential functions, each function having 4 different outputs, means that we need 36 different w parameters.

With this model the decoding (obtained with the command xDecode) gives $[1, 2, 3, 4]$ So in this case the first problem is fixed. On the other hand we calcualte the marginal for the first digit as

$$\mathbb{P}(x_1 = 1|w) = \frac{\overbrace{\phi_1(1)}^{3.6222}}{\underbrace{Z(w)}_{3618.2}} \underbrace{\prod_{j=2}^{4} \sum_{x^j} \phi_j(x^j)}_{383.4795}.$$

By doing so, we get a marginal of 0.3839. In a similar manner we calculate the marginals for the other 3 digits to be one and we obtaine

$$\mathbb{P}(x^2 = 1|w) = 0.2311$$
$$\mathbb{P}(x^3 = 1|w) = 0.0783$$
$$\mathbb{P}(x^4 = 1|w) = 0.0787$$

Comparing with the estimates from the training data given by $0.1530, 0.1949, 0.1471$, we can see that this numbers do not match with the results from the model with untied parameters.

Finally when we calculate the probability for the fourth pin number given that the first three are $1, 2, 3$ we get the following probabilities

```
condNM =

0.1471    0.1013    0.0802    0.1894    0.0731    0.0621    0.0770    0.0733    0.0801    0.1164
```

Even though it predicts well the fact that the most likely value is 4. The model gives a probability of 0.1894, however when we calculate the number of pins that starting with $1, 2, 3$ end up with 4 is 0.9286. Hence the prediction does not give accurate probabilities.

5.
In this case we use the adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

With this the parameter w has 336 components, 36 of them correspond to the unary potentials as in the previous exercises. The 300 more elements comes from the binary potentials where each $\phi_{ij}(x^i, x^j), i < j$ contains 100 hundred possible values and we have 3 potentials, namely $\phi_{12}, \phi_{23}, \phi_{34}$. In this new model, the decoding gives the sequence

$$xDecode = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

To calculate the marginal probability for the first digit to be 1 we need to calculate it as

$$\mathbb{P}(x^1 = 1|w) = \frac{\phi_1(1)}{Z} \sum_{x^2} \phi_2(x^2)\phi_{12}(1, x^2) \overbrace{\sum_{x^3} \phi_{23}(x^2, x^3)\phi_3(x^3) \underbrace{\sum_{x^4} \phi_{34}(x^3, x^4)\phi_4(x^4)}_{M(x^3)}}^{M(x^2)},$$

where we memoize the values of $M(x^2)$ and $M(x^3)$. Similar formulas hold to calculate the marginal for the second third or four digit to be 1. Under this model we have

$$\mathbb{P}(x^1 = 1|w) = 0.3223$$
$$\mathbb{P}(x^2 = 1|w) = 0.3128$$
$$\mathbb{P}(x^3 = 1|w) = 0.1674$$
$$\mathbb{P}(x^4 = 1|w) = 0.1435$$

In this case we can see that the value for the first digit to be one is underestimated and for the second digit to be one is overestimated. But now in this case the marginals for the third and fourth values are pretty close to the value given by the training sample $0.1949, 0.1471$. So we gain accuracy for the last two marginals, but we loose accuracy for the first digit.

Finally under this model we continue to underestimate the value of the fourth digit to be four given that the first three digits are $1, 2, 3$. In this case the probability is given by

$$\mathbb{P}(x^4 = 1 | x^j = j, j = 1, 2, 3) = 0.5028.$$

This estimate is better than the one obtained in part 4 of this exercise, but is not accurate.

6.

2 Bayesian Inference

2.1 Conjugate Priors

Consider a $y \in \{1, 2, 3\}$ following a multinoulli distribution with parameters $\theta = \{\theta_1, \theta_2, \theta_3\}$,

$$y|\theta \sim \text{Mult}(\theta_1, \theta_2, \theta_3).$$

We'll assume that θ follows a Dirichlet distribution (the conjugate prior to the multinoulli) with parameters $\alpha = \{\alpha_1, \alpha_2, \alpha_3\}$,

$$\theta \sim \mathcal{D}(\alpha_1, \alpha_2, \alpha_3).$$

Thus we have

$$p(y|\theta, \alpha) = p(y|\theta) = \theta_1^{I(y=1)}\theta_2^{I(y=2)}\theta_3^{I(y=3)}, \quad p(\theta|\alpha) = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)}\theta_1^{\alpha_1 - 1}\theta_2^{\alpha_2 - 1}\theta_3^{\alpha_3 - 1}.$$

Compute the following quantites:

1. The posterior distribution,
$$p(\theta|y, \alpha).$$

2. The marginal likelihood of y given the hyper-parameters α,
$$p(y|\alpha) = \int p(y, \theta|\alpha) d\theta,$$

3. The posterior mean estimate for θ,
$$\mathbb{E}_{\theta|y,\alpha}[\theta_i] = \int \theta_i p(\theta|y, \alpha) d\theta,$$
which (after some manipulation) should not involve any Γ functions.

4. The posterior predictive distribution for a new independent observation $\hat{y}$ given y,
$$p(\hat{y}|y, \alpha) = \int p(\hat{y}, \theta|y, \alpha) d\theta.$$

Hint: You can use $D(\alpha) = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)}{\Gamma(\alpha_1+\alpha_2+\alpha_3)}$ to represent the normalizing constant of the prior and $D(\alpha^+)$ to give the normalizing constant of the posterior. You will also need to use that $\Gamma(\alpha + 1) = \alpha\Gamma(\alpha)$. For some calculations you may find it a bit cleaner to parameterize the posterior in terms of $\beta_j = I(y = j) + \alpha_j$, and convert back once you have the final result.

Solutions

Before we start with this exercise, let us derive a result that is going to be used frequently. If $\theta|\alpha \sim \mathcal{D}(\alpha_1, \alpha_2, \alpha_3)$. then

$$1 = \int_S \mathbb{P}(\theta|\alpha) d\theta = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)}{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)} \int_S \theta_1^{\alpha_1 - 1}\theta_2^{\alpha_2 - 1}\theta_3^{\alpha_3 - 1} d\theta,$$

where S is the set

$$S = \{(x, y, z) \in \mathbb{R}^3 | 0 \leq x \leq 1, 0 \leq y \leq 1 - x, 0 \leq z \leq 1 - x - y\}.$$

Hence for a, b, c we have

$$\frac{\Gamma(a+b+c+3)}{\Gamma(a+1)\Gamma(b+1)\Gamma(c+1)} = \int_S \theta_1^a \theta_2^b \theta_3^c d\theta. \tag{1}$$

Now we proceed with part 1

1.
We have the model $\alpha \to \theta \to y$, hence $y \perp \alpha | \theta$. Taking this into account and using Bayes rule we have

$$\mathbb{P}(\theta|y,\alpha) \propto \mathbb{P}(y|\theta)\mathbb{P}(\theta|\alpha).$$

By replacing each probability on the RHS by its formula we get

$$\mathbb{P}(\theta|y,\alpha) \propto \prod_{j=1}^{3} \theta^{\beta_j - 1},$$

where the proportionality constant is given by

$$D(\alpha^+) := \int_S \prod_{j=1}^{3} \theta^{\beta_j - 1} d\theta = \frac{\Gamma(\beta_1)\Gamma(\beta_2)\Gamma(\beta_3)}{\Gamma(\beta_1+\beta_2+\beta_3)}.$$

In the last line we used equation (1). With this the full posterior is written as

$$\mathbb{P}(\theta|y,\alpha) = \frac{1}{D(\alpha^+)} \prod_{j=1}^{3} \theta^{\beta_j - 1}. \tag{2}$$

2.
In this case by using the product rule and $y \perp \alpha|\theta$ we have

$$\mathbb{P}(y|\alpha) = \int \mathbb{P}(y,\theta|\alpha) d\theta = \int \mathbb{P}(y|\theta)\mathbb{P}(\theta|\alpha) d\theta.$$

By replacing the appropiate values we end up with

$$\mathbb{P}(y|\alpha) = \frac{1}{D(\alpha)} \int_S \prod_{j=1}^{3} \theta_j^{\beta_j - 1} d\theta.$$

The integral in the RHS is the same integral as in the proportionality constant in the previous exercise, hence we conclude

$$\mathbb{P}(y|\alpha) = \frac{1}{D(\alpha)D(\alpha^+)}.$$

It is not hard to see that $\sum_{y=1}^{3} \mathbb{P}(y|\alpha) = 1$.

3.
Using the posterior obtained in section 1 of this exercise, it is not hard to see that

$$\theta_i \mathbb{P}(\theta|y,\alpha) = \frac{1}{D(\alpha^+)} \prod_{j=1} \theta^{\beta_j - 1 + \delta_{ij}},$$

Hence using equation (1) we get

$$\mathbb{E}[\theta_i] = \int_S \frac{1}{D(\alpha^+)} \prod_{j=1} \theta^{\beta_j - 1 + \delta_{ij}} d\theta = \frac{1}{D(\alpha^+)} \frac{\prod_{j=1}^{3} \Gamma(\beta_j + \delta_{ij})}{\Gamma(\beta_1+\beta_2+\beta_3+1)}.$$

If $\delta_{ij} = 1$ we would like to use the identity $\Gamma(\beta+1) = \Gamma(\beta)$ if $\delta_{ij} = 0$ we don't want to use any identity of the gamma function. This requierment can be written as

$$\Gamma(\beta_j + \delta_{ij}) = \Gamma(\beta_j)(1 + \delta_{ij}(\beta_j - 1).$$

With this we get

$$\mathbb{E}[\theta_i] = \frac{1}{D(\alpha^+)} \frac{\prod_{j=1}^3 \Gamma(\beta_j)(1 + \delta_{ij}(\beta_j - 1))}{\Gamma(\beta_1 + \beta_2 + \beta_3 + 1)}.$$

Recalling the definition of $D(\alpha^+)$ we get

$$\mathbb{E}[\theta_i] = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} \frac{\prod_{j=1}^3 \Gamma(\beta_j)(1 + \delta_{ij}(\beta_j - 1))}{\Gamma(\beta_1 + \beta_2 + \beta_3 + 1)}$$

Finally since $\beta_j = I(y=j) + \alpha_j$ we have $\Gamma(\beta_j) = \Gamma(\alpha_j)(1 + I(y=j)(\alpha_j - 1))$ and $\beta_1 + \beta_2 + \beta_3 = \alpha_1 + \alpha_2 + \alpha_3 + 1$. We finally obtain everything in terms of α

$$\mathbb{E}[\theta_i] = \frac{\Gamma(\alpha_1 + \alpha_2 + \alpha_3)}{\Gamma(\alpha_1)\Gamma(\alpha_2)\Gamma(\alpha_3)} \frac{\prod_{j=1}^3 \Gamma(\alpha_j)(1 + I(y=j)(\alpha_j - 1))(1 + \delta_{ij}(\alpha_j + I(y=j) - 1))}{\Gamma(\alpha_1 + \alpha_2 + \alpha_3 + 2)}$$

By using the recursive property of the Gamma function and simplifying we conclude

$$\mathbb{E}[\theta_i] = \frac{\prod_{j=1}^3 (1 + I(y=j)(\alpha_j - 1))(1 + \delta_{ij}(\alpha_j + I(y=j) - 1))}{(\alpha_1 + \alpha_2 + \alpha_3)(\alpha_1 + \alpha_2 + \alpha_3 + 1)}.$$

simplyfing we conclude

$$\mathbb{E}[\theta_i | y] = \frac{\alpha_y(\alpha_i + 1)}{(\alpha_1 + \alpha_2 + \alpha_3)(\alpha_1 + \alpha_2 + \alpha_3 + 1)}.$$

4.
In this case using independence properties we have

$$\mathbb{P}(\hat{y}|y, \alpha) = \int_S \mathbb{P}(\hat{y}, \theta | y, \alpha) d\theta = \int_S \mathbb{P}(\hat{y}|\theta)\mathbb{P}(\theta|y, \alpha) d\theta.$$

We have a close expression for all the terms in the last integrand in the RHS. By plugging in those expressions we obtain.

$$\mathbb{P}(\hat{y}|y, \alpha) = \frac{1}{D(\alpha^+)} \int_S \prod_{j=1}^3 \theta^{I(\hat{y}=j) + \beta_j - 1} d\theta.$$

Using the result in equation (1) we get

$$\mathbb{P}(\hat{y}|y, \alpha) = \frac{1}{D(\alpha^+)} \frac{\prod_{j=1}^3 \Gamma(I(\hat{y}=j) + \beta_j)}{\Gamma(\beta_1 + \beta_2 + \beta_3 + 1)}.$$

By replacing the value of $D(\alpha^+)$ and using $\Gamma(\beta_j + I(\hat{y}=j)) = \Gamma(\beta_j)(1 + I(\hat{y}=j)(\beta_j - 1))$ we have

$$\mathbb{P}(\hat{y}|y, \alpha) = \frac{\Gamma(\beta_1 + \beta_2 + \beta_3)}{\Gamma(\beta_1)\Gamma(\beta_2)\Gamma(\beta_3)} \frac{\prod_{j=1}^3 \Gamma(\beta_j)(1 + I(\hat{y}=j)(\beta_j - 1))}{\Gamma(\beta_1 + \beta_2 + \beta_3 + 1)} = \frac{\prod_{j=1}^3 (1 + I(\hat{y}=j)(\beta_j - 1))}{\beta_1 + \beta_2 + \beta_3}.$$

Or more concisely

$$\mathbb{P}(\hat{y}|y, \alpha) = \frac{\beta_{\hat{y}}}{\beta_1 + \beta_2 + \beta_3}.$$

2.2 Empirical Bayes

Consider the model

$$y_i \sim \mathcal{N}(w^T \phi(x_i), \sigma^2), \quad w_j \sim \mathcal{N}(0, \lambda).$$

By using properties of Gaussians the marginal likelihood has the form

$$p(y_i | x_i, \sigma, \lambda) = (2\pi)^{-d/2} |C|^{-1/2} \exp\left(-\frac{y^T C^{-1} y}{2}\right),$$

which gives a negative log-marginal likelihood of

$$-\log p(y_i | x_i, \sigma, \lambda) \propto \log |C| + y^T C^{-1} y + \text{const}.$$

where

$$C = \frac{1}{\sigma^2} I + \frac{1}{\lambda} \Phi(X) \Phi(X)^T,$$

As discussed in class, the marginal likelihood can be used to optimize hyper-parameters like σ, λ, and even the basis ϕ.

The demo *example_basis* loads a dataset and fits a degree-2 polynomial to it. Normally we would use a test set to choose the degree of the polynomial but here we'll use the marginal likelihood of the training set. Write a function, *leastSquaresEmpiricalBaysis*, that uses the marginal likelihood to choose the degree of the polynomial as well as the parameters λ and σ (you can assume that all λ_j are equal, and you can restrict your search for λ and σ to powers of 10). Hand in your code and report the marginally most likely values of the degree, σ, and λ. You can use the *logdet* function to compute the log-determinant.

Hint: computing $C^{-1}y$ by explicitly forming C^{-1} may give you numerical issues that lead to non-sensical solution. You can avoid these by using $y^T C^{-1} y = y^T v$ where v is a solution to $Cv = y$ (Matlab will still give a warning due to ill-conditioning, but it won't return non-sensical results).