1 Notes

- Unlike previous courses, it seems like our professor uses (sometimes bold, sometimes not) capitals for vectors, bold capitals for matrices, and unweighted lowercase characters for scalars, but in these notes I use the regular notation where vectors are capital lowercase.

2 Bayes' Theorem and Bayesian Inference

- Bayesian inference refers to the process of calculating the posterior probability using Bayes' Theorem.
- Some mumbo jumbo about the relationships between prior/likelihood and subjective/objective assessment.

3 Simple Bayesian Networks

- Link matrix:

$$\boldsymbol{P}(\boldsymbol{c}|\boldsymbol{p}) = \begin{bmatrix} P(c_1|p_1) & P(c_1|p_2) \\ P(c_2|p_1) & P(c_2|_2) \end{bmatrix} \tag{1}$$

- **Instantiating** a node refers to setting it to a measured value. For instance, if you observe a data point where $c = 1$, you may say c was *instantiated* to 1 for that point.
- **Bayesian classifiers** may introduce more complex tree structures than the *naive* ones.
- Develops intuition for a method that is often intractable.

4 Evidence and Message Passing

- New concepts are necessary to deal with complex networks with intermediate nodes:
  - **Evidence** is information we have at a node – this may be gathered through instantiation (exact values or virtual evidence), or inferred from passing messages. Evidence is unnormalised probabilities so the absolute values are meaningless, but they are useful for making comparisons.
  - **Messages** are information (evidence) passed between nodes to provide evidence to another node.
- This lecture assumes each node has **one** parent.

- **Evidence** (unnormalised probability)
  - π evidence generalises prior probability and comes from parents.
  - λ evidence generalises likelihood and comes from children.
  - Mnemonic: **p**i (π), **p**rior, and "**p**arent" all start with letter "p"; and **l**ambda (λ), **l**ikelihood, and "**l**ad" all start with "l". Further, prior comes *before* so from parents.

- λ **evidence**
  - $\boldsymbol{\lambda(c)} = [\lambda(c_1), \lambda(c_2), \ldots \lambda(c_n)]$
  - $$\lambda(c_i) = \prod_{h=\text{children of } c_i} \sum_{j=\text{indices for } h} \lambda(h_j)P(h_j|c_i) \overset{Vectorised}{=} \boldsymbol{\lambda(h)P(h|c)} \tag{2}$$
  - Note that if any of the leaf notes h_i is not instantiated (i.e. we have no data/evidence on it), we can consider $\lambda(h_i) = 1$ (for all i), thus it's a sum of probabilities which is equal to 1. If a value is known, then only $\lambda(h_j) = 1$ where h_j is the known value, and in the case of virtual evidence, $\lambda(h_j)$ is given by the distribution chosen.
  - If you just want to find the root (hypothesis) node, then you can calculate it just from λ evidence as they all go up.

- π **evidence**
  - This is useful when one wants to calculate the evidence of a node whose parents are not instantiated, but there is some evidence about the state of their parents from other siblings.
  - $$\pi(c_i) = \sum_{p_j=\text{values of parent}} P(c_i|p_j)[\pi(p_j) \prod_{k=\text{children indices of } p_j} \lambda_k(p_j)] \tag{3}$$

- Using λ and π evidence, one may transmit messages up and down the network from the instantiated nodes to find evidence for any node in the network.

- **Notation**
  - $\boldsymbol{\lambda_F(c)}$ means λ evidence from node F for node C.
  - $\boldsymbol{\pi_F(c)}$ means π evidence from *all nodes besides* F for node C.

5 Probability Propagation in Singly Connected Networks

- **Singly connected** means that there is a *single* path between any two nodes in the network.
- The main properties of these networks are:
  - The parents of a node are always independent given their common child (i.e. they don't have a common parent). This lets us calculate their joint probability as the product of their marginals.
  - When updating evidence of a node, the belief propagated through the net to update all nodes is guaranteed to reach a steady state and not loop forever.
- An example of a link matrix for a node with multiple parents looks as follows:

$$\boldsymbol{P(e|w,c)} = \begin{bmatrix} P(e_1|w_1,c_1) & P(e_1|w_1,c_2) & P(e_1|w_2,c_1) & P(e_1|w_2,c_2) \\ P(e_2|w_1,c_1) & P(e_2|w_1,c_2) & P(e_2|w_2,c_1) & P(e_2|w_2,c_2) \\ P(e_3|w_1,c_1) & P(e_3|w_1,c_2) & P(e_3|w_2,c_1) & P(e_3|w_2,c_2) \end{bmatrix} \tag{4}$$

- To calculate the π evidence of a node with 2 parents:

$$\pi(\boldsymbol{E}) = \boldsymbol{P(e|w,c)\pi_e(w,c)} = \boldsymbol{P(e|w,c)\pi_e(w)\pi_e(c)} \tag{5}$$

- To calculate the λ evidence of a node with 2 parents, we have to calculate one λ message for each of the parents. If c has parents a and b, then the evidence from c to a is as follows:

$$\lambda_c(a_i) = \sum_{j=1}^{n} \pi_c \boldsymbol{c}(b_j) \sum_{k=1}^{m} P(c_k|a_i,b)\lambda(c_k) \tag{6}$$

  where n is the number of values that b takes, and m is the number of values c takes.

- **Operations**
  - A and B are the parents of C
  - λ **message** (C to A)

$$\lambda_c(a_i) = \sum_{j=1}^{n} \pi_c \boldsymbol{c}(b_j) \sum_{k=1}^{m} P(c_k|a_i,b)\lambda(c_k) \tag{7}$$

  - π **message** (A to C)

$$\pi_c(a_i) = \begin{cases} 1 & \text{if A is instantiated to } a_k \\ 0 & \text{if A is instantiated but not at } a_k \\ P'(a_i)/\lambda_c(a_i) & \text{if C is not instantiated} \end{cases} \tag{8}$$

- λ **evidence**

$$\lambda(c_k) = \begin{cases} 1 & \text{if C is instantiated to } c_k \\ 0 & \text{if C is instantiated but not at } c_k \\ \prod_i \lambda_{d_i}(c_k) & \text{if C is not instantiated} \end{cases} \tag{9}$$

- π **evidence** (A and B to C)

$$\pi(c_k) = \sum_i \sum_j p(c_k|a_i, b_j)\pi_c(a_i)\pi_c(b_j) \tag{10}$$

- **Posterior probability**

$$P'(c_k) = \alpha\pi(c_k)\lambda(c_k) \tag{11}$$