Probabilistic & Unsupervised Learning: Assignment 1
Callum Lau
Student Number: 19102521

Problem 1.
(a)

Binomial:

$$\begin{aligned}\binom{N}{x}p^x(1-p)^{N-x} =& \binom{N}{x}e^{xlogp+(N-x)log(1-p)}\\ =& \binom{N}{x}(1-p)^N e^{xlog(\frac{p}{1-p})}\end{aligned}$$

$\therefore g(\theta) = (1-p)^N, f(x) = \binom{N}{x}, T(x) = x, \phi(\theta) = log(\frac{p}{1-p})$

Multinomial

$$\begin{aligned}\frac{N!}{x_1!...x_D!}\prod_{d=1}^{D} p_d^{x_d} =& \frac{N!}{x_1!...x_D!}e^{log(\prod_{d=1}^{D} p_d^{x_d})}\\ =& \frac{N!}{x_1!...x_D!}e^{\sum_{d=1}^{D-1} x_d log(p_d)+(N-\sum_{d=1}^{D-1} x_d)log(1-\sum_{i=1}^{D-1} p_i)}\\ &\text{(since we are constrained by} x_D = N - \sum_{d=1}^{D-1} x_d \text{ and } p_D = 1 - \sum_{i=1}^{D-1} p_i \text{)}\\ =& \frac{N!}{x_1!...x_D!}e^{Nlog(1-\sum_{d=1}^{D-1} p_d)}e^{\sum_{d=1}^{D-1} x_d log(\frac{p_d}{p_D})}\\ =& \frac{N!}{x_1!...x_D!}e^{Nlog(1-\sum_{d=1}^{D-1} p_d)}e^{\begin{pmatrix} log(\frac{p_1}{p_D}) & ... & log(\frac{p_{D-1}}{p_D})\end{pmatrix}\begin{pmatrix} x_1 \\ ... \\ x_{D-1}\end{pmatrix}}\end{aligned}$$

$\therefore g(\boldsymbol{\theta}) = (1-\sum_{d=1}^{D-1} p_d)^N, f(\boldsymbol{x}) = \frac{N!}{x_1!...x_D!}, T(\boldsymbol{x}) = \begin{pmatrix} x_1 \\ ... \\ x_{D-1}\end{pmatrix}, \phi_i(\boldsymbol{\theta}) = log(\frac{p_i}{p_D})$

Poisson:

$$\frac{\mu^x e^{-\mu}}{x!} = \frac{e^{-\mu}}{x!}e^{xlog\mu}$$

$\therefore g(\boldsymbol{\theta}) = e^{-\mu}, f(\boldsymbol{x}) = \frac{1}{x!}, T(\boldsymbol{x}) = x, \phi(\boldsymbol{\theta}) = log(\mu)$

Beta:

$$\begin{aligned}\frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1} =&\frac{1}{B(\alpha,\beta)}e^{log(x^{\alpha-1}(1-x)^{\beta-1})}\\ =&\frac{1}{B(\alpha,\beta)}e^{(\alpha-1)logx+(\beta-1)log(1-x)}\\ =&\frac{1}{B(\alpha,\beta)}e^{\begin{pmatrix}\alpha-1 & \beta-1\end{pmatrix}\begin{pmatrix}log(x)\\ log(1-x)\end{pmatrix}}\end{aligned}$$

$\therefore g(\boldsymbol{\theta}) = \frac{1}{B(\alpha,\beta)}, f(\boldsymbol{x}) = 1, T(\boldsymbol{x}) = \begin{pmatrix}\alpha-1\\ \beta-1\end{pmatrix}, \phi(\boldsymbol{\theta}) = \begin{pmatrix}log(x)\\ log(1-x)\end{pmatrix}$

Gamma:

$$\begin{aligned}\frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}e^{-\beta x} =&\frac{\beta^\alpha}{\Gamma(\alpha)}e^{(\alpha-1)log(x)-\beta x}\\ =&\frac{\beta^\alpha}{\Gamma(\alpha)}e^{\begin{pmatrix}\alpha-1 & -\beta\end{pmatrix}\begin{pmatrix}logx\\ x\end{pmatrix}}\end{aligned}$$

$\therefore g(\boldsymbol{\theta}) = \frac{\beta^\alpha}{\Gamma(\alpha)}, f(\boldsymbol{x}) = 1, T(\boldsymbol{x}) = \begin{pmatrix}logx\\ x\end{pmatrix}, \phi(\boldsymbol{\theta}) = \begin{pmatrix}\alpha-1\\ -\beta\end{pmatrix}$

Dirichlet:

$$\begin{aligned}\frac{\Gamma(\sum_{d=1}^D\alpha_d)}{\prod_{d=1}^D\Gamma(\alpha_d)}\prod_{d=1}^D x_d^{\alpha_d-1} =&\frac{\Gamma(\sum_{d=1}^D\alpha_d)}{\prod_{d=1}^D\Gamma(\alpha_d)}e^{log(\prod_{d=1}^D x_d^{\alpha_d-1})}\\ =&\frac{\Gamma(\sum_{d=1}^D\alpha_d)}{\prod_{d=1}^D\Gamma(\alpha_d)}e^{\sum_{d=1}^D(\alpha_d-1)log(x_d)}\\ =&\frac{\Gamma(\sum_{d=1}^D\alpha_d)}{\prod_{d=1}^D\Gamma(\alpha_d)}e^{\begin{pmatrix}\alpha_1-1 & ... & \alpha_D-1\end{pmatrix}\begin{pmatrix}logx_1\\ ...\\ logx_D\end{pmatrix}}\end{aligned}$$

$\therefore g(\boldsymbol{\theta}) = \frac{\Gamma(\sum_{d=1}^D\alpha_d)}{\prod_{d=1}^D\Gamma(\alpha_d)}, f(\boldsymbol{x}) = 1, T(\boldsymbol{x}) = \begin{pmatrix}logx_1\\ ...\\ logx_D\end{pmatrix}, \phi(\boldsymbol{\theta}) = \begin{pmatrix}\alpha_1-1\\ ...\\ \alpha_D-1\end{pmatrix}$

Normal:

$$
\begin{aligned}
\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} =& \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}exp(-\frac{1}{2}(\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x}-\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{x}-\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu}+\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})\\
=& \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}exp(-\frac{1}{2}(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}))exp(-\frac{1}{2}(tr(\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x})-2\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu}))\\
=& \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}exp(-\frac{1}{2}(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}))exp(-\frac{1}{2}(tr(\Sigma^{-1}\boldsymbol{x}\boldsymbol{x}^T)+\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu}))\\
=& \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}exp(-\frac{1}{2}(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}))exp(-\frac{1}{2}vec(\Sigma^{-1})\cdot vec(\boldsymbol{x}\boldsymbol{x}^T)+(\Sigma^{-1}\boldsymbol{\mu})\cdot\boldsymbol{x})
\end{aligned}
$$

where $vec(A)$ = the vectorised form of matrix$A$, and using the dot product.

$$
=\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}exp(-\frac{1}{2}(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu}))exp(\begin{pmatrix}-\frac{1}{2}vec(\Sigma^{-1}) & \Sigma^{-1}\boldsymbol{\mu}\end{pmatrix}\cdot\begin{pmatrix}vec(\boldsymbol{x}\boldsymbol{x}^T)\\ \boldsymbol{x}\end{pmatrix})
$$

$\therefore g(\boldsymbol{\theta}) = |\Sigma|^{-\frac{1}{2}}exp(-\frac{1}{2}(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})), f(\boldsymbol{x}) = (2\pi)^{-\frac{D}{2}}, T(\boldsymbol{x}) = \begin{pmatrix}\boldsymbol{x}\boldsymbol{x}^T\\ \boldsymbol{x}\end{pmatrix}, \phi(\boldsymbol{\theta}) = \begin{pmatrix}-\frac{1}{2}\Sigma^{-1}\\ \Sigma^{-1}\mu\end{pmatrix}$

(b).

Conside a general method used to find $< T(\boldsymbol{x}) >$ for a distribution in exponential family form:

$$
\begin{aligned}
\text{note: } 1 =& \int_{\boldsymbol{x}} g(\boldsymbol{\phi}(\boldsymbol{\theta})f(\boldsymbol{x})e^{\boldsymbol{\phi}(\boldsymbol{\theta})^T T(\boldsymbol{x})}dx = g(\boldsymbol{\phi})\int_{\boldsymbol{x}} f(\boldsymbol{x})e^{\boldsymbol{\phi}^T T(\boldsymbol{x})}d\boldsymbol{x}\\
=> g(\boldsymbol{\phi}) =& \frac{1}{\int_{\boldsymbol{x}} f(\boldsymbol{x})e^{\boldsymbol{\phi}^T T(\boldsymbol{x})}d\boldsymbol{x}}\\
\text{let} A(\boldsymbol{\phi}) :=& -log(g(\boldsymbol{\phi}))\\
\text{then}\frac{\partial A}{\partial \boldsymbol{\phi}_i} =& \frac{\partial}{\partial \boldsymbol{\phi}_i}(log(\int_{\boldsymbol{x}} f(\boldsymbol{x})e^{\boldsymbol{\phi}^T T(\boldsymbol{x})}d\boldsymbol{x}))\\
=& \frac{\int_{\boldsymbol{x}} \frac{\partial}{\partial \boldsymbol{\phi}_i} f(\boldsymbol{x})e^{\boldsymbol{\phi}^T T(\boldsymbol{x})}d\boldsymbol{x}}{\int_{\boldsymbol{x}} f(\boldsymbol{x})e^{\boldsymbol{\phi}^T T(\boldsymbol{x})}d\boldsymbol{x}}\\
=& \frac{\int_{\boldsymbol{x}} f(\boldsymbol{x})T_i(\boldsymbol{x})e^{\boldsymbol{\phi}^T T(\boldsymbol{x})}d\boldsymbol{x}}{(\frac{1}{g(\boldsymbol{\phi})})}\\
=& \int_{\boldsymbol{x}} T_i(\boldsymbol{x})\cdot[f(\boldsymbol{x})g(\boldsymbol{\phi}(\boldsymbol{\theta}))e^{\boldsymbol{\phi}(\boldsymbol{\theta})^T T(\boldsymbol{x})}]d\boldsymbol{x} =< T_i(\boldsymbol{x}) >_{p(x|\theta)}
\end{aligned}
$$

$\therefore$ we can use$\frac{\partial A}{\partial \boldsymbol{\phi}_i} =< T_i(\boldsymbol{x}) >$

Binomial:

$$\boldsymbol{\phi} = log(\frac{p}{1-p}) => p = \frac{e^{\phi}}{1+e^{\phi}}$$

$$\begin{aligned}
A(\boldsymbol{\phi}) = & - log(g(\boldsymbol{\phi})) = -log((1 - \frac{e^{\phi}}{1+e^{\phi}})^{N}) = -Nlog(\frac{1}{1+e^{\phi}}) = Nlog(1+e^{\phi}) \\
\frac{\partial A}{\partial \boldsymbol{\phi}} = & \frac{\partial}{\partial \phi}(-log(g(\boldsymbol{\phi})) = N\frac{\partial}{\partial \phi}(log(1+e^{\phi})) = N(\frac{e^{\phi}}{1+e^{\phi}}) \\
= & N(\frac{e^{log(\frac{p}{1-p})}}{1+e^{log(\frac{p}{1-p})}}) = N(\frac{\frac{p}{1-p}}{1+\frac{p}{1-p}}) \\
= & Np = < x > = < T(x) >_{p(x|\theta)}
\end{aligned}$$

$=$

Multinomial:

$$\begin{aligned}
\phi_i(\boldsymbol{\theta}) = & log(\frac{p_i}{p_D}) => p_i = e^{\phi_i} p_D \\
\text{note that} \sum_{i=1}^{D} p_i = & 1 => p_D \sum_{i=1}^{D} e^{\phi_i} = 1 \\
=> p_D = & \frac{1}{\sum_{i=1}^{D} e^{\phi_i}} \\
\therefore p_i = & \frac{e^{\phi_i}}{\sum_{d=1}^{D} e^{\phi_d}} \\
\therefore \frac{\partial A(\boldsymbol{\phi})}{\partial \phi_i} = & \frac{\partial}{\partial \phi_i}(-Nlog(1 - \sum_{d=1}^{D-1} p_d) = \frac{\partial}{\partial \phi_i}(-Nlog(1 - \sum_{d=1}^{D-1} p_d)) \\
= & N\frac{\partial}{\partial \phi_i}(-log(1 - \sum_{d=1}^{D-1} \frac{e^{\phi_d}}{\sum_{j=1}^{D} e^{\phi_d}})) \\
= & N\frac{\partial}{\partial \phi_i}(-log(1 - \sum_{d=1}^{D-1} \frac{e^{\phi_d}}{\sum_{j=1}^{D} e^{\phi_j}})) \\
= & N\frac{\partial}{\partial \phi_i}(log(\sum_{d=1}^{D} e^{\phi_d})) \text{ (dropping constant term)} \\
= & N\frac{e^{\phi_i}}{\sum_{d=1}^{D} e^{\phi_d}} = Np_i = < x_i >
\end{aligned}$$

Poisson:

$$
\begin{aligned}
< x >=& \sum_{x=0}^{\infty} x \frac{\mu^x e^{-\mu}}{x!} = \sum_{x=1}^{\infty} \frac{\mu^x e^{-\mu}}{(x-1)!} = \mu e^{-\mu} \sum_{x=1}^{\infty} \frac{\mu^{x-1}}{(x-1)!} \\
=& \mu e^{-\mu} \sum_{k=0}^{\infty} \frac{\mu^k}{(k)!} = \mu e^{-\mu} e^{\mu} = \mu = e^{\phi}
\end{aligned}
$$

Beta:

$$
\begin{aligned}
\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} &= \begin{pmatrix} \alpha - 1 \\ \beta - 1 \end{pmatrix} => \alpha = \phi_1 + 1, \beta = \phi_2 + 1 \\
A(\boldsymbol{\phi}) &= -\, log(g(\boldsymbol{\phi})) = -log(B(\phi_1 + 1, \phi_2 + 1)) \\
\frac{\partial A}{\partial \phi_1} =& \frac{B'(\phi_1 + 1, \phi_2 + 1)}{B(\phi_1 + 1, \phi_2 + 1)} = \frac{B'(\alpha, \beta)}{B(\alpha, \beta)} =< log(x) > \\
\frac{\partial A}{\partial \phi_2} =& \frac{B'(\phi_1 + 1, \phi_2 + 1)}{B(\phi_1 + 1, \phi_2 + 1)} = \frac{B'(\alpha, \beta)}{B(\alpha, \beta)} =< log(1 - x) >
\end{aligned}
$$

Gamma:

$$
\begin{aligned}
\begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} &= \begin{pmatrix} \alpha - 1 \\ -\beta \end{pmatrix} => \alpha = \phi_1 + 1, \beta = -\phi_2 \\
A(\boldsymbol{\phi}) &= -\, log(g(\phi_1, \phi_2) = -log(\frac{(-\phi_2)^{\phi_1 + 1}}{\Gamma(\phi_1 + 1)}) = log(\Gamma(\phi_1 + 1) - (\phi_1 + 1) log(-\phi_2) \\
\frac{\partial A}{\partial \phi_1} =& \frac{\Gamma'(\phi_1 + 1)}{\Gamma(\phi_1 + 1)} - log(-\phi_2) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)} - log(\beta) =< log(x) > \\
\frac{\partial A}{\partial \phi_2} =& \frac{(\phi_1 + 1)}{-\phi_2} = \frac{\alpha}{\beta} =< x >
\end{aligned}
$$

Dirichlet:

$$
\begin{aligned}
\begin{pmatrix} \phi_1 \\ ... \\ \phi_D \end{pmatrix} &= \begin{pmatrix} \alpha_1 - 1 \\ ... \\ \alpha_D - 1 \end{pmatrix} => \begin{pmatrix} \alpha_1 \\ ... \\ \alpha_D \end{pmatrix} = \begin{pmatrix} \phi_1 + 1 \\ ... \\ \phi_D + 1 \end{pmatrix} \\
g(\boldsymbol{\theta}) =& \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)}{\prod_{d=1}^{D} \Gamma(\alpha_d)} => g(\boldsymbol{\phi}) = \frac{\Gamma(\sum_{d=1}^{D} (\phi_d + 1))}{\prod_{d=1}^{D} \Gamma(\phi_d + 1)} \\
\frac{\partial A(\boldsymbol{\phi})}{\partial \phi_i} =& \frac{\partial}{\partial \phi_i}(-log(g(\phi_1, ..., \phi_D)) = \frac{\partial}{\partial \phi_i} \sum_{d=1}^{D} log(\Gamma(\phi_d + 1)) - \frac{\partial}{\partial \phi_i} log(\Gamma(\sum_{d=1}^{D} (\phi_d + 1)))
\end{aligned}
$$

$$= -\frac{\Gamma'(\sum_{d=1}^{D}(\phi_d+1))}{\Gamma(\sum_{d=1}^{D}(\phi_d+1))} + \frac{\Gamma'(\phi_i+1)}{\Gamma(\phi_i+1)} = -\frac{\Gamma'(\sum_{d=1}^{D}\alpha_d)}{\Gamma(\sum_{d=1}^{D}\alpha_d)} + \frac{\Gamma'(\alpha_i)}{\Gamma(\alpha_i)} =< log(x_i) >$$

Normal:

$$\begin{aligned}
\begin{pmatrix}\phi_1\\ \phi_2\end{pmatrix} &= \begin{pmatrix}-\frac{1}{2}\Sigma^{-1}\\ \Sigma^{-1}\mu\end{pmatrix} => \begin{pmatrix}\Sigma\\ \mu\end{pmatrix} = \begin{pmatrix}-\frac{1}{2}\phi_1^{-1}\\ -\frac{1}{2}\phi_1^{-1}\phi_2\end{pmatrix}\\
A(\boldsymbol{\phi}) &= -\,log(g(\boldsymbol{\phi}(\boldsymbol{\theta})) = -log(|\Sigma|^{-\frac{1}{2}}exp(-\frac{1}{2}(\boldsymbol{\mu}^T\Sigma^{-1}\boldsymbol{\mu})))\\
&= -\frac{1}{2}log(|-2\phi_1|) - \frac{1}{4}(\phi_1^{-1}\phi_2)^T\phi_1(\phi_1^{-1}\phi_2)\\
&= -\frac{1}{2}log(|-2\phi_1|) - \frac{1}{4}\phi_2^T\phi_1^{-1}\phi_2 = -\frac{1}{2}log(-2^D|\phi_1|) - \frac{1}{4}\phi_2^T\phi_1^{-1}\phi_2\\
=> \frac{\partial A}{\partial\phi_1} &= -\frac{1}{2}\times\frac{|\phi_1|(\phi_1^{-1})^T}{|\phi_1|} - \frac{1}{4}\times\phi_1^{-T}\phi_2\phi_2^T\phi_1^{-T}\\
&= -\frac{1}{2}\phi_1^{-1} - \frac{1}{4}\phi_1^{-1}\phi_2\phi_2^T\phi_1^{-1}
\end{aligned}$$

using the fact that the inverse transpose of a symmetric matrix is the same as the inverse

$$\begin{aligned}
&=\Sigma - \mu\mu^T =< \boldsymbol{x}\boldsymbol{x}^T >\\
\frac{\partial A}{\partial\phi_2} &=\frac{\partial}{\partial\phi_2}(-\frac{1}{2}log(-2^D|\phi_1|) - \frac{1}{4}\phi_2^T\phi_1^{-1}\phi_2)\\
&= -\frac{1}{4}\times 2\phi^{-1}\phi_2\\
&= -\frac{1}{2}\phi_1^{-1}\phi_2\\
&=\mu =< \boldsymbol{x} >
\end{aligned}$$

Problem 2.)

Consider i.i.d samples $\boldsymbol{x_i, i = 1, ..., n}$drawn from an exponential family distribution, and noting we can write $\phi = \phi(\theta)$then:

$$p(\boldsymbol{x}|\boldsymbol{\phi}) =g(\boldsymbol{\phi})f(\boldsymbol{x})e^{\boldsymbol{\phi}^T T(\boldsymbol{x})} => p(\boldsymbol{x}_i|\boldsymbol{\phi}) = g(\boldsymbol{\phi})f(\boldsymbol{x}_i)e^{\boldsymbol{\phi}^T T(\boldsymbol{x}_i)}$$

then the log-likelihood function takes the form

$$\begin{aligned}
l(\boldsymbol{\phi}) &=log(\prod_{i=1}^{n} g(\boldsymbol{\phi})f(\boldsymbol{x}_i)e^{\boldsymbol{\phi}^T T(\boldsymbol{x}_i)})\\
&=\sum_{i=1}^{n} log(g(\boldsymbol{\phi})) + \sum_{i=1}^{n} log(f(\boldsymbol{x_i})) + \sum_{i=1}^{n}\boldsymbol{\phi}^T T(\boldsymbol{x_i})\\
=> \frac{\partial l}{\partial\boldsymbol{\phi}} &=N\frac{\nabla g(\boldsymbol{\phi})}{g(\boldsymbol{\phi})} + \sum_{i=1}^{n} T(\boldsymbol{x}_i) =^! 0
\end{aligned}$$

$$=> \frac{1}{N}\sum_{i=1}^{n} T(\boldsymbol{x}_i) = -\frac{\nabla g(\hat{\boldsymbol{\phi}})}{g(\hat{\boldsymbol{\phi}})} = \frac{\partial}{\partial \hat{\boldsymbol{\phi}}}(-log(g(\hat{\boldsymbol{\phi}}))) =: \frac{\partial}{\partial \hat{\boldsymbol{\phi}}} A(\hat{\boldsymbol{\phi}}) = E[T(\boldsymbol{x})]_{\phi}$$

where the last equality follows from the general result proved in 1b.)

$$\text{(equivalently)}\ \frac{1}{N}\sum_{i=1}^{n} T_k(\boldsymbol{x}_i) = E[T_k(x)]$$

Problem 3

(a)

Each pixel $x_d^{(n)}$ has a support of only discrete and binary values $\{0, 1\}$, however the Gaussian distribution has a continuous support $(-\infty, \infty)$, with negative values, suggesting that for all the pixels, a multivariate Gaussian would not be appropriate. Also, the dimensionality of the multivariate Gaussian would be the same as the length as the column vectors D, which would mean for D large (even D=64), we would have to work with a Gaussian with a (DxD) covariance matrix. This would pose an significant computational problem when trying to invert the covariance matrix, as well as finding its determinant, in order to produce the multivariate Gaussian equation. (A multivariate bernoulli would be much easier to work with)

(b)

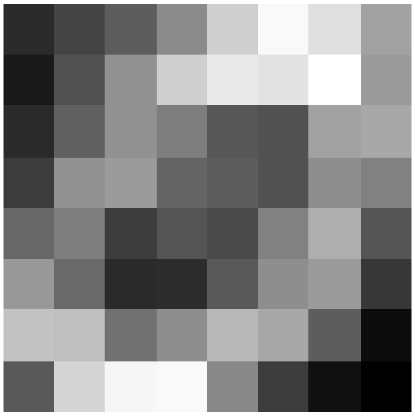
$$\begin{aligned}
P(x|p) &= \prod_{d=1}^{D} p_d^{x_d}(1-p_d)^{(1-x_d)} \\
=> l(p) &= log(\prod_{n=1}^{N}\prod_{d=1}^{D} p_d^{x_d}(1-p_d)^{(1-x_d)}) = \sum_{n=1}^{N}\sum_{d=1}^{D} log(p_d^{x_d^{(n)}}(1-p_d)^{(1-x_d^{(n)})}) \\
&= \sum_{n=1}^{N}\sum_{d=1}^{D}(x_d^{(n)} log(p_d) + (1-x_d^{(n)}) log(1-p_d)) \\
\frac{\partial l}{\partial p_i} &= \sum_{n=1}^{N}(\frac{x_i^{(n)}}{p_i} - \frac{1-x_i^{(n)}}{1-p_i}) \overset{!}{=} 0 \\
=> (1-\hat{p}_i)\sum_{n=1}^{N} x_i^{(n)} &= \hat{p}_i \sum_{n=1}^{N}(1-x_i^{(n)}) \\
\hat{p}_i &= \frac{1}{N}\sum_{n=1}^{N} x_i^{(n)} \\
=> \hat{\boldsymbol{p}} &= (\frac{1}{N}\sum_{n=1}^{N} x_1^{(n)} ... \frac{1}{N}\sum_{n=1}^{N} x_D^{(n)})^T
\end{aligned}$$

(c)

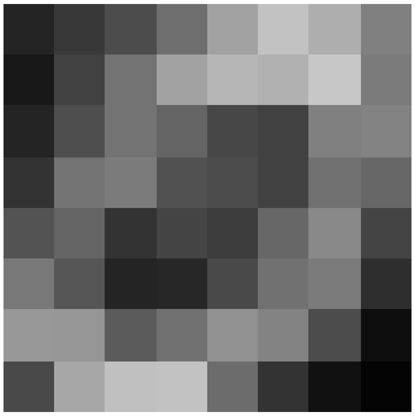
$$
\begin{aligned}
P(p|x) &\propto P(x|p)P(p) \\
=> log(P(p|x) &\propto log(\prod_{n=1}^{N}\prod_{d=1}^{D} p_d^{x_d}(1-p_d)^{(1-x_d)} \prod_{d=1}^{D} \frac{p_d^{\alpha-1}(1-p_d)^{\beta-1}}{B(\alpha,\beta)}) \\
&= \sum_{n=1}^{N}\sum_{d=1}^{D}(x_d^{(n)} log(p_d) + (1-x_d^{(n)}) log(1-p_d)) + \sum_{d=1}^{D}((\alpha-1) log p_d + (\beta-1) log(1-p_d) \\
=> \frac{\partial}{\partial p_i} log(P(p|x) &= \sum_{n=1}^{N}(\frac{x_i^{(n)}}{p_i} - \frac{1-x_i^{(n)}}{1-p_i}) + \frac{\alpha-1}{p_i} - \frac{\beta-1}{1-p_i} \overset{!}{=} 0 \\
=> (1-\hat{p_i})[\sum_{n=1}^{N} x_i^{(n)} + \alpha - 1] &= \hat{p_i}[\sum_{n=1}^{N}(1-x_i^{(n)}) + \beta - 1] \\
[\sum_{n=1}^{N} x_i^{(n)} + \alpha - 1] &= \hat{p_i}[\sum_{n=1}^{N} x_i^{(n)} - \sum_{n=1}^{N} x_i^{(n)} + \sum_{n=1}^{N}(1) + \alpha - 1 + \beta - 1] = \hat{p_i}[N + \alpha + \beta - 2] \\
\hat{p_i} &= \frac{\sum_{n=1}^{N} x_i^{(n)} + \alpha - 1}{N + \alpha + \beta - 2} \\
=> \hat{\boldsymbol{p}} &= (\frac{\sum_{n=1}^{N} x_1^{(n)} + \alpha - 1}{N + \alpha + \beta - 2} ... \frac{\sum_{n=1}^{N} x_D^{(n)} + \alpha - 1}{N + \alpha + \beta - 2})^T
\end{aligned}
$$

(d) The python code for importing the binary txt data set and displaying the parameter images for both the MLE and MAP estimates is given below

```
import numpy as np
from matplotlib import pyplot as plt

def main():
    # load the data set
    Y = np.loadtxt('binarydigits.txt')
    N, D = Y.shape
    muML = Y.sum(axis=0)/N
    mlImg = np.reshape(muML, (8,8))

    plt.figure()
    plt.imshow(mlImg,
               interpolation="None",
               cmap='gray',
               vmin=0., vmax=1.)
    plt.axis('off')
    plt.show()

    alpha = 3
    beta = 3
    muMAP = (Y.sum(axis=0) + alpha - 1)/(N + alpha + beta - 2)
    mapImg = np.reshape(muMAP, (8,8))

    plt.figure()
    plt.imshow(mapImg,
               interpolation="None",
               cmap='gray',
               vmin=0., vmax=1.)
    plt.axis('off')
    plt.show()

if __name__ == "__main__":
    main()
```



(e)



For both plots we have the values at each pixel shown by a greyscale colour scheme, where for the p_i

estimates 0 = black, 1 = white. For the MAP estimate we see from the image (or more clearly from the values themselves) that all values become more 'centred' around the value 0.5 (for all p_i), since the prior for p_i is given by the Beta distribution with mean $\frac{\alpha}{\alpha+\beta}$ is 0.5 in the case $\alpha = \beta$ (note in the case $\alpha = \beta, \alpha$ large we have that $P(p_i|x) \approx 0.5$).The MAP estimate may be a better estimator than ML when we are sure that we have an informative prior. However, in this case and seeing the whole data set, we see that each image resembles either 0, 5 or 7. Therefore a prior which gives greater weighting to the probability of black or white in any particular pixel as being equal is certainly uninformative/wrong (since the "mean" image will form some shape corresponding to a mix of 0,5,7 which all have black spots near the center and corners). Hence the MAP estimate is worse than the ML estimate. Finally, note that for small α, β values, the effect on the posterior is somewhat small, so we can't say that the MAP estimate is *considerably* worse.

Problem 4
Using the fact that the marginal likelihood given for a model is $P(D|M_i) = \int d\theta_i P(D|\theta_i, M_i)P(\theta_i|M_i)$
(a)

$$\begin{aligned}
P(D|M_1) =& \prod_{n=1}^{N}\prod_{d=1}^{D} 0.5^{x_d^{(n)}}(1-0.5)^{1-x_d^{(n)}} \\
=& \prod_{n=1}^{N}\prod_{d=1}^{D} 0.5^{x_d^{(n)}} 0.5^{1-x_d^{(n)}} = \prod_{n=1}^{N} 0.5^{D} \\
=& (0.5)^{ND} \\
=> log(D|M_1) =& NDlog(0.5)
\end{aligned}$$

(b)
Since $p \sim U[0,1]$, then $P(p|M_2) = 1$

$$\begin{aligned}
P(D|M_2) =& \int_0^1 \prod_{n=1}^{N} P(x_n|p, M_2)P(p|M_2)dp = \int_0^1 \{\prod_{n=1}^{N}\prod_{d=1}^{D} p^{x_d^{(n)}}(1-p)^{1-x_d^{(n)}}\} \times 1dp \\
=& \int_0^1 \prod_{n=1}^{N} p^{\sum_{d=1}^{D} x_d^{(n)}}(1-p)^{D-\sum_{d=1}^{D} x_d^{(n)}} dp \\
=& \int_0^1 p^{\sum_{n=1}^{N}\sum_{d=1}^{D} x_d^{(n)}}(1-p)^{ND-\sum_{n=1}^{N}\sum_{d=1}^{D} x_d^{(n)}} dp \\
=& B(\sum_{n=1}^{N}\sum_{d=1}^{D} x_d^{(n)} + 1, ND + 1 - \sum_{n=1}^{N}\sum_{d=1}^{D} x_d^{(n)}) \\
=> log(D|M_2) =& logB(\sum_{n=1}^{N}\sum_{d=1}^{D} x_d^{(n)} + 1, ND + 1 - \sum_{n=1}^{N}\sum_{d=1}^{D} x_d^{(n)})
\end{aligned}$$

where B is the Beta function, since the integral of the Beta Distribution over its support is equal to 1.

(c)
Since $p_d \sim U[0,1]$,then $P(p_d|M_3) = 1$

$$\begin{aligned}
P(D|M_3) &= \int_0^1 \cdots \int \{\prod_{n=1}^{N} \prod_{d=1}^{D} p_d^{x_d^{(n)}} (1-p_d)^{1-x_d^{(n)}}\} \times 1 \times ... \times 1 dp_1 ... dp_D \\
&= \int_0^1 \cdots \int \prod_{d=1}^{D} p_d^{\sum_{n=1}^{N} x_d^{(n)}} (1-p_d)^{N-\prod_{n=1}^{N} x_d^{(n)}} dp_1 ... dp_D \\
&= \int_0^1 p_1^{\sum_{n=1}^{N} x_1^{(n)}} (1-p_1)^{N-\sum_{n=1}^{N} x_1^{(n)}} dp_1 ... \int_0^1 p_D^{\sum_{n=1}^{N} x_D^{(n)}} (1-p_D)^{N-\sum_{n=1}^{N} x_D^{(n)}} dp_D \\
&= B(\sum_{n=1}^{N} x_1^{(n)} + 1, N + 1 - \sum_{n=1}^{N} x_1^{(n)}) ... B(\sum_{n=1}^{N} x_D^{(n)} + 1, N + 1 - \sum_{n=1}^{N} x_D^{(n)}) \\
&= \prod_{d=1}^{D} B(\sum_{n=1}^{N} x_d^{(n)} + 1, N + 1 - \sum_{n=1}^{N} x_d^{(n)}) \\
=> log(P(D|M_3) &= \sum_{d=1}^{D} log(B(\sum_{n=1}^{N} x_d^{(n)} + 1, N + 1 - \sum_{n=1}^{N} x_d^{(n)}))
\end{aligned}$$

Now note that

$$\begin{aligned}
P(M_i|D) &= \frac{P(D|M_i)P(M_i)}{P(D)} = \frac{P(D|M_i) \times \frac{1}{3}}{P(D|M_1) \times \frac{1}{3} + P(D|M_2) \times \frac{1}{3} + P(D|M_3) \times \frac{1}{3}} \\
&= \frac{P(D|M_i)}{P(D|M_1) + P(D|M_2) + P(D|M_3)}
\end{aligned}$$

and each posterior is$P(M_i|D) \propto P(D|M_i)P(M_i)$

$$log(P(M_i|D) \propto log(P(D|M_i)) + log(P(M_i))$$

After plugging in the $x_d^{(n)}$ values for all of the equations we find:

$$\begin{aligned}
logP(M_1|D) &\approx -4436 + log(1/3) = -4437 => P(M_1|D) \approx e^{-4437} \\
logP(M_2|D) &\approx -4284 + log(1/3) = -4285 => P(M_2|D) \approx e^{-4285} \\
logP(M_3|D) &\approx -3851 + log(1/3) = -3852 => P(M_3|D) \approx e^{-3852}
\end{aligned}$$

Therefore, comparing the relative posterior probabilities the Bayesian model (Model 3) is significantly more likely to be correct.

Problem 5

(a) Consider the data set $X = \{x_1, x_2, .., x_N\}, x_i \in R^D$of examination marks for N different exams, with D students taking each exam. For each exam we would expect different mean **and** variances in the marks obtained. This is because different examiners will be giving the different exams, e.g. for a hard exam you'd expect to see high variance. This means that factor analysis would be appropriate if we wanted to reduce the dimensions of this data set. The underlying factors could then be two unobserved variables: logical and creative skill. We could then make the assumption that each examination score is linearly related to these two factors. This is a reasonable model since, for high logical skill we would expect better scores on science exams - maths, physics, chemistry etc, and for high creative skill better scores on arts exams- english, history, languages. Using factor analysis we could then analyse how the variability for each exam is related to the variability in logical and creative aptitude. One possible conclusion would be that logical aptitude accounts for the majority of the variability in exam scores, or vice versa. The Gaussianity assumption would certainly be reasonable for a high number of students (D large) since exams marks are usually distributed normally by design. However, the linearity assumption does not seem reasonable, since it is unlikely that there is a direct tradeoff in exam performance, when logical and creative skill vary. We could modify this model by introducing an extra latent, corresponding to another unobserved 'skill' that would influence the exam scores.

(b) Consider the problem of modelling a data set of the heights of a population. A valid assumption to make is that these heights are normally distributed. However, within the total population we would expect variations in the mean and variance of the height distribution for different ethinicities. So each mixture component would correspond to another Gaussian (with different mean and variance) for each ethnicity within the population (the height within each ethnicity would also be normally distributed). The number of clusters would be equal to the number of ethnicities we choose to include. For example, if we were modelling the height of the world population, we could include Asian, Indo-Asian, African, Caucasian and Latin as 5 components/clusters of the total population (since these make up broadly the vast majority of the world ethnicities). We could then select the weightings for each component according to the proportion of that component/ethnicity within the world population. This would be a sensible model for the real world, since it is evident that different ethnicities have at least a large difference in their mean height, and each ethnicity will certainly have its own normal distribution, as well as the underlying world population.

Problem 6

a)

$$p(x) = \int p(x|z)p(z)dz$$

since this corresponds to the product of two Gaussians, it is again a Gaussian given by

$$\begin{aligned} p(x) \sim & N(E[x], Cov[x]) \\ \text{where} E_x[x] =& E_z[E_{x|z}(\Lambda z)] = E_z(\Lambda z) = \Lambda E[z] = 0 \\ Cov_x[x] =& E_z[Cov[x|z]] + Cov_z[E[x|z]] \\ =& E_z(\Psi I) + Cov_z(\Lambda z) \\ =& \Psi I + \Lambda \Upsilon \Lambda^T \end{aligned}$$

Let Υ be the diagonal matrix of the eigenvalues of some matrix Q. Then since Λ, Λ^T are orthonormal then$\Lambda \Upsilon \Lambda^T$ can be written as:

$QQ^T = \Lambda \mathit{\Upsilon} \Lambda^T$for some arbitrary, not-necessarily-orthonormal, matrix Q - i.e. the eigendecomposition of QQ^T(which is real and symmetric). Hence

$$p(x) \sim N(0, \Psi I + QQ^T) \text{ for some arbitrary Q}$$

And therefore the alternative form can model the exact same set of marginal distributions since $p(x)$ takes the same form in the standard model (as derived in lecture slides):

$$p(x) = N(0, \Psi I + \Lambda\Lambda^T)$$

(b)

Consider x fixed. Then:

$$\begin{aligned}
p(z|x) \propto & p(z)p(x|z) \\
= & c \times exp(-\frac{1}{2}(z^T \mathit{\Upsilon}^{-1} z)) \times exp(-\frac{1}{2}(x - \Lambda z)^T \Psi^{-1}(x - \Lambda z)) \text{ (for some constant c)} \\
= & c \times exp(-\frac{1}{2}\{z^T \mathit{\Upsilon}^{-1} z + (x - \Lambda z)^T \Psi^{-1}(x - \Lambda z)\}) \\
= & c \times exp(-\frac{1}{2}\{z^T \mathit{\Upsilon}^{-1} z + \Psi^{-1} x^T x - 2\Psi^{-1} z^T \Lambda^T x + \Psi^{-1} z^T \Lambda^T \Lambda z\}) \\
= & c' \times exp(-\frac{1}{2}\{z^T \mathit{\Upsilon}^{-1} z - 2\Psi^{-1} z^T \Lambda^T x + \Psi^{-1} z^T I z\}) \text{ (for some constant c', and using } \Lambda^T\Lambda = I) \\
= & c' \times exp(-\frac{1}{2}\{z^T (\mathit{\Upsilon}^{-1} + \Psi^{-1} I) z - 2\Psi^{-1} z^T \Lambda^T x\}) \\
= & c'' \times exp(-\frac{1}{2}\{z^T \Sigma^{-1} z - 2z^T \Sigma^{-1}\mu + \mu^T \Sigma^{-1} \mu\}) = c'' exp(-\frac{1}{2}(z - \mu)^T \Sigma^{-1}(z - \mu)
\end{aligned}$$

where $\Sigma = (\Upsilon^{-1} + \Psi^{-1} I)^{-1}$ and $\mu = \Psi^{-1}\Sigma\Lambda^T x$

hence $p(z|x) \sim N(z|\Psi^{-1}\Sigma\Lambda^T x, (\Upsilon^{-1} + \Psi^{-1} I)^{-1}) = N(\mu, \Sigma)$

Now consider the mean and variance in the limit $\Psi \to 0$, we recover the PCA $p(z|x)$:

$$\begin{aligned}
\lim_{\Phi \to 0} \Sigma = \lim_{\Phi \to 0} \Psi^{-1}(\mathit{\Upsilon}^{-1} + \Psi^{-1} I)^{-1} &= \lim_{\Phi \to 0} (\begin{pmatrix} \frac{1}{\Upsilon_1} & \cdots & 0 \\ \dots & \ddots & \dots \\ 0 & \cdots & \frac{1}{\Upsilon_K} \end{pmatrix} + \begin{pmatrix} \frac{1}{\Psi} & \cdots & 0 \\ \dots & \ddots & \dots \\ 0 & \cdots & \frac{1}{\Psi} \end{pmatrix})^{-1} \\
&= \lim_{\Phi \to 0} (\begin{pmatrix} \frac{\Psi + \Upsilon_1}{\Psi \Upsilon_1} & \cdots & 0 \\ \dots & \ddots & \dots \\ 0 & \cdots & \frac{\Psi + \Upsilon_K}{\Psi \Upsilon_K} \end{pmatrix})^{-1} = \lim_{\Phi \to 0} \begin{pmatrix} \frac{\Psi \Upsilon_1}{\Psi + \Upsilon_1} & \cdots & 0 \\ \dots & \ddots & \dots \\ 0 & \cdots & \frac{\Psi \Upsilon_K}{\Psi + \Upsilon_K} \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 \\ \dots & \ddots & \dots \\ 0 & \cdots & 0 \end{pmatrix}
\end{aligned}$$

which is what we expect since PCA is non-probabilistic.

$$\lim_{\Phi \to 0} \mu = \lim_{\Phi \to 0} \Psi^{-1}\Sigma\Lambda^T x$$

$$= \lim_{\Phi\to 0} \frac{1}{\Psi} \begin{pmatrix} \frac{\Psi\Upsilon_1}{\Psi+\Upsilon_1} & \cdots & 0 \\ \cdots & \ddots & \cdots \\ 0 & \cdots & \frac{\Psi\Upsilon_K}{\Psi+\Upsilon_K} \end{pmatrix} \Lambda^T x$$

$$= \lim_{\Phi\to 0} \begin{pmatrix} \frac{\Upsilon_1}{\Psi+\Upsilon_1} & \cdots & 0 \\ \cdots & \ddots & \cdots \\ 0 & \cdots & \frac{\Upsilon_K}{\Psi+\Upsilon_K} \end{pmatrix} \Lambda^T x = I\Lambda^T x = \Lambda^T x$$

hence $\lim\limits_{\Phi\to 0} \mu = \Lambda^T x$

Problem 7

Beliefs are inconsistent since:

$$b(A) = 0.5, b(A\bigcap B) = 0.5, b(B) = 0.6, b(A\bigcup B) = 0.7$$

consider by the basic lawsof probability beliefs should obey:

$$b(A\bigcup B) = b(A) + b(B) - b(A\bigcap B)$$

but $b(A) + b(B) - b(A\bigcap B) = 0.5 + 0.6 - 0.5 = 0.6 \neq 0.7 = b(A\bigcup B)$

Construct a Dutch book as follows:

$$\left\{\begin{array}{ll} Belief & probability \\ b(A) & = 0.5 \\ b(A\bigcap B) & = 0.5 \\ b(B) & = 0.6 \\ b(A\bigcup B) & = 0.7 \end{array}\right. \quad \overset{\text{he is willing to accept bets on}}{=>} \quad \left\{\begin{array}{ll} Event & \text{win:loss} \\ notA & \text{at } 5:5 \\ A\bigcap B & \text{at } 5:5 \\ notB & \text{at } 6:4 \\ A\bigcup B & \text{at } 3:7 \end{array}\right.$$

Therefore, for all possible event combinations, his winnings are:

$$\left\{\begin{array}{l} Event \\ A\bigcap B \\ not(A)\bigcap B \\ A\bigcap not(B) \\ not(A)\bigcap not(B) \end{array}\right. \quad \overset{\text{he wins}}{=>} \left\{\begin{array}{llllll} \text{notA} & \text{A}\bigcap\text{B} & \text{notB} & \text{A}\bigcup\text{B} & & winnings \\ -5 & +5 & -4 & +3 & = & -1 \\ +5 & -5 & -4 & +3 & = & -1 \\ -5 & -5 & +6 & +3 & = & -1 \\ +5 & -5 & +6 & -7 & = & -1 \end{array}\right.$$

Hence according to his beliefs, he will lose money not matter the outcome.

Problem 8
(a)

$$\begin{aligned}
L = \sum_{n=1}^{N} ||\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n||^2 &= \sum_{n=1}^{N} ||QP\boldsymbol{x}_n - \boldsymbol{x}_n||^2 \\
&= \sum_{n=1}^{N} (QP\boldsymbol{x}_n - \boldsymbol{x}_n)^T (QP\boldsymbol{x}_n - \boldsymbol{x}_n) \\
(*) &= \sum_{n=1}^{N} (tr(\boldsymbol{x}_n^T P^T Q^T QP\boldsymbol{x}_n) - \boldsymbol{x}_n^T P^T Q^T \boldsymbol{x}_n - \boldsymbol{x}_n^T PQ\boldsymbol{x}_n + \boldsymbol{x}_n^T \boldsymbol{x}) \\
(*) => \frac{\partial L}{\partial Q} &= \sum_{n=1}^{N} (2Q(P\boldsymbol{x}_n)(P\boldsymbol{x}_n)^T - (\boldsymbol{x}_n\boldsymbol{x}_n^T)P^T - (\boldsymbol{x}_n\boldsymbol{x}_n^T)P^T) \overset{!}{=} 0 \\
\sum_{n=1}^{N} Q(P\boldsymbol{x}_n)(P\boldsymbol{x}_n)^T &= (\sum_{n=1}^{N} \boldsymbol{x}_n\boldsymbol{x}_n^T)P^T \\
QP(\sum_{n=1}^{N} \boldsymbol{x}_n\boldsymbol{x}_n^T)P^T &= (\sum_{n=1}^{N} \boldsymbol{x}_n\boldsymbol{x}_n^T)P^T \\
QP\sum\nolimits_x P^T &= \sum\nolimits_x P^T \\
Q &= \sum\nolimits_x P^T (P\sum\nolimits_x P^T)^{-1} \\
(*) => \frac{\partial L}{\partial P} &= \sum_{n=1}^{N} (Q^TQ)^T P\boldsymbol{x}_n\boldsymbol{x}_n^T + Q^TQP\boldsymbol{x}_n\boldsymbol{x}_n^T - 2Q^T\boldsymbol{x}_n\boldsymbol{x}_n^T \overset{!}{=} 0 \\
2Q^TQP\sum_{n=1}^{N} \boldsymbol{x}_n\boldsymbol{x}_n^T &= 2Q^T \sum_{n=1}^{N} \boldsymbol{x}_n\boldsymbol{x}_n^T \\
P &= (Q^TQ)^{-1}Q^T
\end{aligned}$$

(proofs use various matrix differentiation formulae found in the Matrix Cookbook)
P and Q give a minimum since L is a convex function

(b)

$$\begin{aligned}
Q_*^T Q_* &= (QC^{-1})^T (QC^{-1}) \\
&= C^{-T} Q^T Q C^{-1}
\end{aligned}$$

note that $Q^TQ$is (KxK)and real, symmetric by construction, therefore, we can eigendecompose Q^TQ as

$Q^TQ = ADA^T$ where A is orthogonal with columns the eigenvectors of Q^TQ
and D diagonal with the eigenvalues of Q^TQ

$$\begin{aligned}
=> Q_*^T Q_* =& C^{-T} A D A^T C^{-1} \text{ hence choose } C = D^{\frac{1}{2}} A^T \\
=& (D^{\frac{1}{2}} A^T)^{-T} A D A (D^{\frac{1}{2}} A^T)^{-1} \\
=& (D^{-\frac{1}{2}})^T A^{-1} A D A^T A^{-T} D^{-\frac{1}{2}} \\
=& D^{-\frac{1}{2}} D D^{-\frac{1}{2}} = I
\end{aligned}$$

And so, using the identity found in (a) we have:

$$\begin{aligned}
P =& (Q^T Q)^{-1} Q^T \\
=> C^{-1} P_* =& ((Q_* C)^T (Q_* C))^{-1} (Q_* C)^T \\
P_* =& C (C^T Q_*^T Q_* C)^{-1} C^T Q_*^T \\
=& C (C^T C)^{-1} C^T Q_*^T \\
=& C C^{-1} C^{-T} C^T Q_*^T = Q_*^T
\end{aligned}$$

(c)

$$\begin{aligned}
min \sum_n ||\hat{x}_n - x_n|| =& min \sum_n (Q P x_n - x_n)^T (Q P x_n - x_n) \\
\text{since } Q = Q_* C \text{ and } P = C^{-1} Q_*^T & \\
=& min \sum_n (Q_* C C^{-1} Q_*^T x_n - x_n)^T (Q_* C C^{-1} Q_*^T x_n - x_n)^T \\
=& min \sum_n (Q_* Q_*^T x_n - x_n)^T (Q_* Q_*^T x_n - x_n) \\
=& min \sum_n \{x_n^T Q_* Q_*^T Q_* Q_*^T x_n - 2 x_n^T Q_* Q_*^T x_n + x_n^T x_n\} \\
=& min \sum_n \{x_n^T x_n - x_n^T Q_* Q_*^T x_n\} \\
\text{since } Q_*^T Q_* =& I \text{ and disregarding the constant term } \sum_n \{x_n^T x_n\} \\
=& min \sum_n \{-x_n^T Q_* Q_*^T x_n\} = max \sum_n \{x_n^T Q_* Q_*^T x_n\} \\
=& max \sum_n \{Tr(x_n^T Q_* Q_*^T x_n)\} \text{ (trace(scalar)=scalar)} \\
=& max \sum_n \{Tr(Q Q_*^T x_n x_n^T)\} \text{ (cyclic property of trace)} \\
=& max Tr(Q_* Q_*^T \sum_n x_n x_n^T) \text{ (Tr(A)+Tr(B)=Tr(A+B))} \\
=& max Tr(Q_* Q_*^T \Sigma_x) \\
=& max Tr(Q_*^T \Sigma_x Q_*) = (*)
\end{aligned}$$

Consider Σ_x as the covariance matrix of the input vectors, since the input distribution has zero mean. Then $Q_*^T \Sigma_x Q_*$ is a projection of the covariance matrix from (DxD) dimensional space to (KxK) dimensional space. Note that when we take the trace of a covariance matrix, we are taking the sum of the variances. Therefore, in the projection we want to maximise the variances that appear on the K diagonals. We know from PCA that these are precisely the K largest eigenvalues of the covariance matrix. Hence, in choosing Q_*, we should select the matrix which 'diagonalises' (not quite diagonalise as the dimension is reduced) Σ_x such that the largest K eigenvalues appear on the diagonal. The columns of Q_* therefore must consist of the K eigenvectors corresponding to the largest K eigenvectors of Σ_x. Since we can write for $Q_* = (v_1 ... v_K)$ (ignoring the proportion constants), such that $\Sigma_x v_i = \lambda_i v_i$:

$$\begin{aligned} tr(Q_*^T \Sigma_x Q_*) =& tr(\begin{matrix} v_1 & \cdots & v_K \end{matrix})^T \Sigma_x (\begin{matrix} v_1 & \cdots & v_K \end{matrix}) \\ =& tr(\begin{matrix} v_1 & \cdots & v_K \end{matrix})^T (\begin{matrix} \lambda_1 v_1 & \cdots & \lambda_K v_K \end{matrix}) \\ =& tr(\begin{matrix} \lambda_1 v_1^T v_1 & \cdots & \lambda_K v_K^T v_K \end{matrix}) \; = \sum_{i=1}^{K} \lambda_i \text{ where } |v_i| = 1 \end{aligned}$$

(d)
In the case the uniquenesses are known then we have:

$$\begin{aligned} z_k =& \sum_i P_{xi} x_i \\ \hat{x}_j =& \sum_k Q_{jk} z_k + \epsilon_j \\ \text{with } \epsilon_j \sim& N(0, \Psi_{jj}) \end{aligned}$$

Since e_j have known covariance, the identities found in (a) and (b) for P, Q remain the same, hence the minimisation problem becomes:

$$\begin{aligned} min \sum_{n=1}^{N} ||\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n||^2 =& min \sum_{n=1}^{N} ||QP\boldsymbol{x}_n + \epsilon_n - \boldsymbol{x}_n||^2 \\ =& min \sum_n (Q_* Q_*^T x_n + (\epsilon_n - x_n))^T (Q_* Q_*^T x_n + (\epsilon_n - x_n)) \\ =& min \sum_n \{x_n^T Q_* Q_*^T Q_* Q_*^T x_n - x_n^T Q_* Q_*^T (x_n - \epsilon_n) - (x_n^T - \epsilon_n^T) Q_* Q_*^T x_n + (x_n^T - \epsilon_n^T)(x_n - \epsilon_n)\} \\ =& min \sum_n \{-x_n^T Q_* Q_*^T x_n + x_n^T Q_* Q_*^T \epsilon_n + \epsilon_n^T Q_* Q_*^T x_n\} \\ =& min \sum_n \{2 x_n^T Q_* Q_*^T \epsilon_n - x_n^T Q_* Q_*^T x_n\} = min \sum_n \{x_n^T Q_* Q_*^T (2\epsilon_n - x_n)\} \end{aligned}$$

Which can then be minimised by finding Q_*

(e)
It is possible to use an autoencoder for unknown uniqueness if you remove the linearity condition, since it could then learn the dependencies between the latents.