Solution Manual-Machine Learning A Probabilistic Perspective (Kevin P. Murphy)

Machine Learning (National University of Singapore)

Machine Learning: A Probabilistic Perspective

Solutions Manual
(Please do not make publicly available)

Kevin P. Murphy

The MIT Press
Cambridge, Massachusetts
London, England

Chapter 1

Introduction

1.1 Solutions

1.1.1 KNN classifier on shuffled MNIST data

We just have to insert the following piece of code.

Listing 1.1: Part of `mnistShuffled1NNdemo`

```
... load data

%% permute columns
D = 28*28;
setSeed(0); perm = randperm(D);
Xtrain = Xtrain(:, perm);
Xtest = Xtest(:, perm);

... same as before
```

1.1.2 Approximate KNN classifiers

According to John Chia, the following code will work.

Listing 1.2: :

```
[result, ndists] = flann_search(Xtrain', Xtest', 1, ...
  struct('algorithm', 'kdtree', 'trees', 8, 'checks', 64));
errorRate = mean(ytrain(result) ~= ytest0)
```

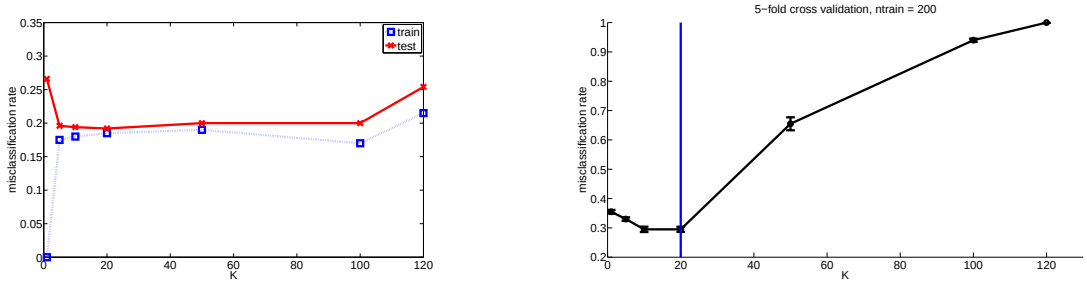
He reports the following results on MNIST with 1NN.

	ntests=1000		ntests=10,000	
	Err	Time	Err	Time
Flann	4.8%	17s	3.35%	17.2s
Vanilla	3.8%	3.68s	3.09%	28.36s

So the approximate method is somewhat faster for large test sets, but is slightly less accurate.

1.1.3 CV for KNN

See Figure 1.1(b). The CV estimate is an overestimate of the test error, but has the right shape. Note, however, that the empirical test error is only based on 500 test points. A better comparison would use a much larger test set.



(a) (b)

Figure 1.1: (a) Misclassification rate vs K in a K-nearest neighbor classifier. On the left, where K is small, the model is complex and hence we overfit. On the right, where K is large, the model is simple and we underfit. Dotted blue line: training set (size 200). Solid red line: test set (size 500). (b) 5-fold cross validation estimate of test error. Figure generated by `knnClassifyDemo`.

Chapter 2

Probability

2.1 Solutions

2.1.1 Probabilities are sensitive to the form of the question that was used to generate the answer

1. The event space is shown below, where X is one child and Y the other.

X	Y	Prob.
G	G	1/4
G	B	1/4
B	G	1/4
B	B	1/4

Let N_g be the number of girls and N_b the number of boys. We have the constraint (side information) that $N_b + N_g = 2$ and $0 \leq N_b, N_g \leq 2$. We are told $N_b \geq 1$ and are asked to compute the probability of the event $N_g = 1$ (i.e., one child is a girl). By Bayes rule we have

$$p(N_g = 1|N_b \geq 1) = \frac{p(N_b \geq 1|N_g = 1)p(N_g = 1)}{p(N_b \geq 1)} \tag{2.1}$$

$$= \frac{1 \times 1/2}{3/4} = 2/3 \tag{2.2}$$

2. Let Y be the identity of the observed child and X be the identity of the other child. We want $p(X = g|Y = b)$. By Bayes rule we have

$$p(X = g|y = b) = \frac{p(Y = b|X = g)p(X = g)}{p(Y = b)} \tag{2.3}$$

$$= \frac{(1/2) \times (1/2)}{1/2} = 1/2 \tag{2.4}$$

Tom Minka (Minka 1998) has written the following about these results:

> This seems like a paradox because it seems that in both cases we could condition on the fact that "at least one child is a boy." But that is not correct; you must condition on the event actually observed, not its logical implications. In the first case, the event was "He said yes to my question." In the second case, the event was "One child appeared in front of me." The generating distribution is different for the two events. Probabilities reflect the number of possible ways an event can happen, like the number of roads to a town. Logical implications are further down the road and may be reached in more ways, through different towns. The different number of ways changes the probability.

2.1.2 Legal reasoning

Let E be the evidence (the observed blood type), and I be the event that the defendant is innocent, and $G = \neg I$ be the event that the defendant is guilty.

1. The prosecutor is confusing $p(E|I)$ with $p(I|E)$. We are told that $p(E|I) = 0.01$ but the relevant quantity is $p(I|E)$. By Bayes rule, this is

$$p(I|E) = \frac{p(E|I)p(I)}{p(E|I)p(I) + p(E|G)p(G)} = \frac{0.01p(I)}{0.01p(I) + (1 - p(I))} \tag{2.5}$$

since $p(E|G) = 1$ and $p(G) = 1 - p(I)$. So we cannot determine $p(I|E)$ without knowing the prior probability $p(I)$. So $p(E|I) = p(I|E)$ only if $p(G) = p(I) = 0.5$, which is hardly a presumption of innocence.

To understand this more intuitively, consider the following isomorphic problem (from http://en.wikipedia.org/wiki/Prosecutor's_fallacy):

> A big bowl is filled with a large but unknown number of balls. Some of the balls are made of wood, and some of them are made of plastic. Of the wooden balls, 100 are white; out of the plastic balls, 99 are red and only 1 are white. A ball is pulled out at random, and observed to be white.

Without knowledge of the relative proportions of wooden and plastic balls, we cannot tell how likely it is that the ball is wooden. If the number of plastic balls is far larger than the number of wooden balls, for instance, then a white ball pulled from the bowl at random is far more likely to be a white plastic ball than a white wooden ball — even though white plastic balls are a minority of the whole set of plastic balls.

2. The defender is quoting $p(G|E)$ while ignoring $p(G)$. The prior odds are

$$\frac{p(G)}{p(I)} = \frac{1}{799,999} \tag{2.6}$$

The posterior odds are

$$\frac{p(G|E)}{p(I|E)} = \frac{1}{7999} \tag{2.7}$$

So the evidence has increased the odds of guilt by a factor of 1000. This is clearly relevant, although perhaps still not enough to find the suspect guilty.

2.1.3 Variance of a sum

We have

$$\begin{aligned}
\text{var}\left[X+Y\right] &= E[(X+Y)^2] - (E[X]+E[Y])^2 && (2.8)\\
&= E[X^2+Y^2+2XY] - (E[X]^2+E[Y]^2+2E[X]E[Y]) && (2.9)\\
&= E[X^2] - E[X]^2 + E[Y^2] - E[Y]^2 + 2E[XY] - 2E[X]E[Y] && (2.10)\\
&= \text{var}\left[X\right] + \text{var}\left[Y\right] + 2\text{cov}\left[X,Y\right] && (2.11)
\end{aligned}$$

If X and Y are independent, then $\text{cov}\left[X,Y\right] = 0$, so $\text{var}\left[X+Y\right] = \text{var}\left[X\right] + \text{var}\left[Y\right]$.

2.1.4 Bayes rule for medical diagnosis

Let $T = 1$ represent a positive test outcome, $T = 0$ represent a negative test outcome, $D = 1$ mean you have the disease, and $D = 0$ mean you don't have the disease. We are told

$$\begin{aligned}
P(T=1|D=1) &= 0.99 && (2.12)\\
P(T=0|D=0) &= 0.99 && (2.13)\\
P(D=1) &= 0.0001 && (2.14)
\end{aligned}$$

We are asked to compute $P(D=1|T=1)$, which we can do using Bayes' rule:

$$\begin{aligned}
P(D=1|T=1) &= \frac{P(T=1|D=1)P(D=1)}{P(T=1|D=1)P(D=1)+P(T=1|D=0)P(D=0)} && (2.15)\\
&= \frac{0.99\times 0.0001}{0.99\times 0.0001 + 0.01\times 0.9999} && (2.16)\\
&= 0.009804 && (2.17)
\end{aligned}$$

So although you are much more likely to have the disease (given that you have tested positive) than a random member of the population, you are still unlikely to have it.

2.1.5 The Monty Hall problem

Let H_i denote the hypothesis that the prize is behind door i. We make the following assumptions: the three hypotheses H_1, H_2 and H_3 are equiprobable *a priori*, i.e.,,

$$P(H_1) = P(H_2) = P(H_3) = \frac{1}{3}. \tag{2.18}$$

The datum we receive, after choosing door 1, is one of $D = 3$ and $D = 2$ (meaning door 3 or 2 is opened, respectively). We assume that these two possible outcomes have the following probabilities. If the prize is behind door 1 then the host has a free choice; in this case we assume that the host selects at random between $D = 2$ and $D = 3$. Otherwise the choice of the host is forced and the probabilities are 0 and 1.

$$\left| \begin{array}{l} P(D=2\|H_1)=\frac{1}{2} \\ P(D=3\|H_1)=\frac{1}{2} \end{array} \right| \left| \begin{array}{l} P(D=2\|H_2)=0 \\ P(D=3\|H_2)=1 \end{array} \right| \left| \begin{array}{l} P(D=2\|H_3)=1 \\ P(D=3\|H_3)=0 \end{array} \right| \tag{2.19}$$

Now, using Bayes theorem, we evaluate the posterior probabilities of the hypotheses:

$$P(H_i\|D=3) = \frac{P(D=3\|H_i)P(H_i)}{P(D=3)} \tag{2.20}$$

$$\left| \; P(H_1\|D=3) = \tfrac{(1/2)(1/3)}{P(D=3)} \; \right| \; P(H_2\|D=3) = \tfrac{(1)(1/3)}{P(D=3)} \; \left| \; P(H_3\|D=3) = \tfrac{(0)(1/3)}{P(D=3)} \; \right| \tag{2.21}$$

The denominator $P(D = 3)$ is $(1/2)$ because it is the normalizing constant for this posterior distribution. So

$$\left| \; P(H_1\|D=3) \;=\; \frac{1}{3} \; \right| \; P(H_2\|D=3) \;=\; \frac{2}{3} \; \left| \; P(H_3\|D=3) \;=\; 0. \; \right| \tag{2.22}$$

So the contestant should switch to door 2 in order to have the biggest chance of getting the prize.

Many people find this outcome surprising. There are two ways to make it more intuitive. One is to play the gamethirty times with a friend and keep track of the frequency with which switching gets the prize. Alternatively, you can perform a thought experiment in which the game is played with a million doors. The rules are now that the contestant chooses one door, then the game show host opens 999,998 doors in such a way as not to reveal the prize, leaving the *contestant's* selected door and *one other door* closed. The contestant may now stick or switch. Imagine the contestant confronted by a million doors, of which doors 1 and 234,598 have not been opened, door 1 having been the contestant's initial guess. Where do you think the prize is?

Another way to think about the problem is to use a directed graphical model of the form $P \rightarrow M \leftarrow F$, where P indicates the location the prize, F indicates your first choice, and M indicates which door Monty opens. Clearly P and F cause (determine) M. When we observe M, our belief about P changes because we have observed evidence about its child M.

2.1.6 Moments of a Bernoulli distribution

Mean

$$\mathbb{E}[X] = \sum_{x \in \{0,1\}} xp(x) = 0p(X=0) + 1p(X=1) = \theta \tag{2.23}$$

Variance

$$\begin{aligned} \mathrm{var}[X] &= \mathbb{E}\left[(X-\mu)^2\right] = \sum_{x \in \{0,1\}} p(x)(x-\mu)^2 && (2.24) \\ &= \theta(1-\theta)^2 + (1-\theta)(0-\theta)^2 && (2.25) \\ &= \theta(1+\theta^2-2\theta) + (1-\theta)\theta^2 && (2.26) \\ &= \theta + \theta^3 - 2\theta^2 + \theta^2 - \theta^3 && (2.27) \\ &= \theta - \theta^2 = \theta(1-\theta) && (2.28) \end{aligned}$$

Alternative proof

$$\begin{aligned} \mathbb{E}\left[X^2\right] &= 0^2 p(x=0) + 1^2 p(x=1) = \theta && (2.29) \\ \mathrm{var}[X] &= \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 = \theta - \theta^2 = \theta(1-\theta) && (2.30) \end{aligned}$$

2.1.7 Conditional independence

1. Bayes' rule gives

$$P(H|E_1, E_2) = \frac{P(E_1, E_2|H)P(H)}{P(E_1, E_2)} \tag{2.31}$$

Thus the information in (ii) is sufficient. In fact, we don't need $P(E_1, E_2)$ because it is equal to the normalization constant (to enforce the sum to one constraint). (i) and (iii) are insufficient.

2. Now the equation simplifies to

$$P(H|E_1, E_2) = \frac{P(E_1|H)P(E_2|H)P(H)}{P(E_1, E_2)} \tag{2.32}$$

so (i) and (ii) are obviously sufficient. (iii) is also sufficient, because we can compute $P(E_1, E_2)$ using normalization.

2.1.8 Pairwise independence does not imply mutual independence

We provide two counter examples.

Let X_1 and X_2 be independent binary random variables, and $X_3 = X_1 \oplus X_2$, where $\oplus$ is the XOR operator. We have $p(X_3|X_1, X_2) \neq p(X_3)$, since X_3 can be deterministically calculated from X_1 and X_2. So the variables $\{X_1, X_2, X_3\}$ are not mutually independent. However, we also have $p(X_3|X_1) = p(X_3)$, since without X_2, no information can be provided to X_3. So $X_1 \perp X_3$ and similarly $X_2 \perp X_3$. Hence $\{X_1, X_2, X_3\}$ are pairwise independent.

Here is a different example. Let there be four balls in a bag, numbered 1 to 4. Suppose we draw one at random. Define 3 events as follows:

- X_1: ball 1 or 2 is drawn.
- X_2: ball 2 or 3 is drawn.
- X_3: ball 1 or 3 is drawn.

We have $p(X_1) = p(X_2) = p(X_3) = 0.5$. Also, $p(X_1, X_2) = p(X_2, X_3) = p(X_1, X_3) = 0.25$. Hence $p(X_1, X_2) = p(X_1)p(X_2)$, and similarly for the other pairs. Hence the events are pairwise independent. However, $p(X_1, X_2, X_3) = 0 \neq 1/8 = p(X_1)p(X_2)p(X_3)$.

2.1.9 Conditional independence iff joint factorizes

Independency $\Rightarrow$ Factorization. Let $g(x, z) = p(x|z)$ and $h(y, z) = p(y|z)$. If $X \perp Y|Z$ then

$$p(x, y|z) = p(x|z)p(y|z) = g(x, z)h(y, z) \tag{2.33}$$

Factorization $\Rightarrow$ Independency. If $p(x, y|z) = g(x, z)h(y, z)$ then

$$1 = \sum_{x,y} p(x, y|z) = \sum_{x,y} g(x, z)h(y, z) = \sum_x g(x, z) \sum_y h(y, z) \tag{2.34}$$

$$p(x|z) = \sum_y p(x, y|z) = \sum_y g(x, z)h(y, z) = g(x, z) \sum_y h(y, z) \tag{2.35}$$

$$p(y|z) = \sum_x p(x, y|z) = \sum_x g(x, z)h(y, z) = h(y, z) \sum_x g(x, z) \tag{2.36}$$

$$p(x|z)p(y|z) = g(x, z)h(y, z) \sum_x g(x, z) \sum_y g(y, z) \tag{2.37}$$

$$= g(x, z)h(y, z) = p(x, y|z) \tag{2.38}$$

2.1.10 Conditional independence

1. True, since

$$(X \perp W|Z, Y) \Rightarrow p(X|W, Z, Y) = p(X|Z, Y) \tag{2.39}$$

$$(X \perp Y|Z) \Rightarrow p(X|Z, Y) = p(X|Z) \tag{2.40}$$

$$\Rightarrow p(X|W, Z, Y) = p(X|Z) \tag{2.41}$$

$$\Rightarrow (X \perp Y, W|Z) \tag{2.42}$$

2. False. Consider the DAG in Figure 2.1. It encodes that $(X \perp Y|Z)$ and $(X \perp Y|W)$ but not $(X \perp Y|Z, W)$.

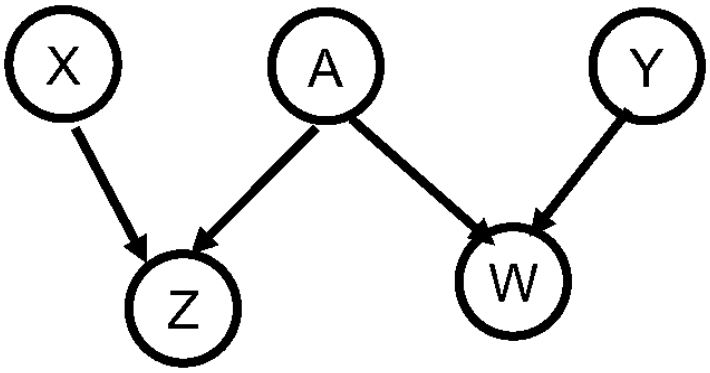


Figure 2.1: A DGM.

2.1.11 Deriving the inverse gamma density

We have

$$p_y(y) = p_x(x)|\frac{dx}{dy}| \tag{2.43}$$

where

$$\frac{dx}{dy} = -\frac{1}{y^2} = -x^2 \tag{2.44}$$

So

$$\begin{aligned} p_y(y) &= x^2 \frac{b^a}{\Gamma(a)} x^{a-1} e^{-xb} && (2.45)\\ &= \frac{b^a}{\Gamma(a)} x^{a+1} e^{-xb} && (2.46)\\ &= \frac{b^a}{\Gamma(a)} y^{-(a+1)} e^{-b/y} = \mathrm{IG}(y|a,b) && (2.47) \end{aligned}$$

2.1.12 Normalization constant for a 1D Gaussian

Following the first hint we have

$$\begin{aligned} Z^2 &= \int_0^{2\pi}\int_0^{\infty} r \exp\left(-\frac{r^2}{2\sigma^2}\right) dr d\theta && (2.48)\\ &= \left[\int_0^{2\pi} d\theta\right]\left[\int_0^{\infty} r \exp\left(-\frac{r^2}{2\sigma^2}\right) dr\right] && (2.49)\\ &= (2\pi) I && (2.50) \end{aligned}$$

where I is the inner integral

$$I = \int_0^{\infty} r \exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{2.51}$$

Following the second hint we have

$$\begin{aligned} I &= -\sigma^2 \int -\frac{r}{\sigma^2} e^{-r^2/2\sigma^2} dr && (2.52)\\ &= -\sigma^2 \left[e^{-r^2/2\sigma^2}\right]_0^{\infty} && (2.53)\\ &= -\sigma^2[0-1] = \sigma^2 && (2.54) \end{aligned}$$

Hence

$$\begin{aligned} Z^2 &= 2\pi\sigma^2 && (2.55)\\ Z &= \sigma\sqrt{(2\pi)} && (2.56) \end{aligned}$$

2.1.13 Expressing mutual information in terms of entropies

$$
\begin{aligned}
I(X,Y) &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} && (2.57)\\
&= \sum_{x,y} p(x,y) \log \frac{p(x|y)}{p(x)} && (2.58)\\
&= -\sum_{x,y} p(x,y) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) && (2.59)\\
&= -\sum_{x} p(x) \log p(x) - \left(-\sum_{x,y} p(x,y) \log p(x|y) \right) && (2.60)\\
&= -\sum_{x} p(x) \log p(x) - \left(-\sum_{y} p(y) \sum_{x} p(x|y) \log p(x|y) \right) && (2.61)\\
&= H(X) - H(X|Y) && (2.62)
\end{aligned}
$$

We can show $I(X,Y) = H(Y) - H(Y|X)$ by symmetry.

2.1.14 Mutual information for correlated normals

The entropy is

$$h(X,Y) = \frac{1}{2} \log \left[(2\pi e)^2 \det \Sigma \right] = \frac{1}{2} \log \left[(2\pi e)^2 \sigma^4 (1-\rho^2) \right] \tag{2.63}$$

Since X and Y are individually normal with variance σ^2, we have

$$h(X) = h(Y) = \frac{1}{2} \log \left[2\pi e \sigma^2 \right] \tag{2.64}$$

Hence

$$
\begin{aligned}
I(X,Y) &= h(X) + h(Y) - h(X,Y) && (2.65)\\
&= \log[2\pi e\sigma^2] - \frac{1}{2} \log[(2\pi e)^2 \sigma^4 (1-\rho^2)] && (2.66)\\
&= \frac{1}{2} \log[(2\pi e\sigma^2)^2] - \frac{1}{2} \log[(2\pi e^2 \sigma^2)^2 (1-\rho^2)] && (2.67)\\
&= \frac{1}{2} \log \frac{1}{1-\rho^2} = -\frac{1}{2} \log[1-\rho^2] && (2.68)
\end{aligned}
$$

1. $\rho = 1$. In this case, $X = Y$, and $I(X,Y) = \infty$, which makes sense.

2. $\rho = 0$. In this case, X and Y are independent, and $I(X,Y) = 0$, which makes sense.

3. $\rho = -1$. In this case, $X = -Y$, and $I(X,Y) = \infty$, which again makes sense.

2.1.15 A measure of correlation (normalized mutual information)

1. We have

$$r = \frac{H(X) - H(Y|X)}{H(X)} = \frac{H(Y) - H(Y|X)}{H(X)} = \frac{I(X,Y)}{H(X)} \tag{2.69}$$

where the second step follows since $H(X) = H(Y)$

2. Since $0 \leq H(Y|X) \leq H(Y) = H(X)$ we have

$$0 \leq \frac{H(Y|X)}{H(X)} \leq 1 \tag{2.70}$$

$$0 \geq -\frac{H(Y|X)}{H(X)} \geq -1 \tag{2.71}$$

$$1 \geq 1 - \frac{H(Y|X)}{H(X)} \geq 0 \tag{2.72}$$

$$1 \geq \frac{H(X) - H(Y|X)}{H(X)} \geq 0 \tag{2.73}$$

$$1 \geq r \geq 0 \tag{2.74}$$

3. $r = 0$ iff $I(X, Y) = 0$ iff X and Y are independent.

4. $r = 1$ iff $H(Y|X) = 0$ iff Y is a deterministic function of X. By symmetry, $r = 1$ iff $H(X|Y) = 0$ iff X is a deterministic function of Y. Hence X and Y must have a one-to-one relationship.

2.1.16 MLE minimizes KL divergence to the empirical distribution

We want to compute

$$q^* = \operatorname*{argmin}_q \mathbb{KL}(p||q) = \operatorname*{argmin}_q \sum_{\mathbf{x}} p(x) \log \frac{p(x)}{q(x)} \tag{2.75}$$

We can drop the $\sum_{\mathbf{x}} p(x) \log p(x)$ term, which is independent of q, yielding the following cross entropy objective:

$$q^* = \arg\min_q -\sum_{\mathbf{x}} p(x) \log q(x) \tag{2.76}$$

Now suppose p is the empirical distribution, which puts a probability atom on the observed training data and zero mass everywhere else:

$$p_{emp}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_i}(x) \tag{2.77}$$

Using the sifting property of delta functions we get

$$\mathbb{KL}(p_{emp}||q) = \text{const} - \sum_{\mathbf{x}} p_{emp}(x) \log q(x) \tag{2.78}$$

$$= \text{const} - \sum_{\mathbf{x}} \left[\frac{1}{N} \sum_i \delta_{x_i}(x) \right] \log q(x) \tag{2.79}$$

$$= \text{const} - \frac{1}{N} \sum_i \log q(x_i) \tag{2.80}$$

This is just the average negative log likelihood of q on the training set.

2.1.17 Mean, mode, variance for the beta distribution

For the mode we can use simple calculus, just as in Section **??**, to show that

$$\text{mode}[\boldsymbol{\theta}] = \frac{a-1}{a+b-2} \tag{2.81}$$

For the mean we have

$$\mathbb{E}[\theta|\mathcal{D}] = \int_0^1 \theta p(\theta|\mathcal{D}) d\theta \tag{2.82}$$

$$= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int \theta^{(a+1)-1}(1-\theta)^{b-1} d\theta \tag{2.83}$$

$$= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+1)\Gamma(b)}{\Gamma(a+1+b)} = \frac{a}{a+b} \tag{2.84}$$

where we used the definition of the Gamma function (Equation **??**) and the fact that $\Gamma(x+1) = x\Gamma(x)$.

We can find the variance in the same way, by first showing that

$$\mathbb{E}\left[\theta^2\right] = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^1 \frac{\Gamma(a+2+b)}{\Gamma(a+2)\Gamma(b)} \theta^{(a+2)-1}(1-\theta)^{b-1} d\theta \tag{2.85}$$

$$= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+2+b)}{\Gamma(a+2)\Gamma(b)} = \frac{a}{a+b} \frac{a+1}{a+1+b} \tag{2.86}$$

Now we use $\mathrm{var}[\theta] = \mathbb{E}\left[\theta^2\right] - \mathbb{E}[\theta]^2$ and $\mathbb{E}[\theta] = a/(a+b)$ to get the variance.

2.1.18 Expected value of the minimum of two rv's

Let $Z = \min(X, Y)$. We have $P(Z > z) = P(X > z, Y > z) = P(X > z)p(Y > z) = (1-z)^2 = 1 + z^2 - 2z$. Hence the cdf of the minimum is $P(Z \leq z) = 2z - z^2$ and the pdf is $p(z) = 2 - 2z$. Hence the expected value is

$$\mathbb{E}[Z] = \int_0^1 z(2-2z) = 1/3 \tag{2.87}$$

Chapter 3

Generative models for discrete data

3.1 Solutions

3.1.1 MLE for the Bernoulli/ binomial model

The log-likelihood is

$$\ell(\theta) = \sum_{i=1}^{N} \log \mathrm{Ber}(x_i|\theta) = \sum_{i=1}^{N} \log \left[\theta^{x_i}(1-\theta)^{1-x_i}\right] = N_1 \log\theta + N_2 \log(1-\theta) \tag{3.1}$$

where $N_1 = \sum_i x_i$ is the number of heads and $N_2 = \sum_i (1-x_i)$ is the number of tails. To find the MLE, we find the maximum of this expression as follows:

$$\frac{d\ell}{d\theta} = \frac{N_1}{\theta} - \frac{N_2}{1-\theta} = 0 \tag{3.2}$$

$$N_1 = \hat{\theta}(N_2 + N_1) \tag{3.3}$$

$$\hat{\theta} = \frac{N_1}{N_1 + N_2} \tag{3.4}$$

where $N_1 + N_2 = N$.

3.1.2 Marginal likelihood for the Beta-Bernoulli model

For integers,

$$(\alpha)(\alpha+1)\cdots(\alpha+M-1) \tag{3.5}$$

$$= \frac{(\alpha+M-1)!}{(\alpha-1)!} \tag{3.6}$$

$$= \frac{(\alpha+M-1)(\alpha+M-2)\cdots(\alpha+M-M)(\alpha+M-M-1)\cdots 2\cdot 1}{(\alpha-1)(\alpha-2)\cdots 2\cdot 1} \tag{3.7}$$

$$= \frac{(\alpha+M-1)(\alpha+M-2)\cdots(\alpha)(\alpha-1)\cdots 2\cdot 1}{(\alpha-1)(\alpha-2)\cdots 2\cdot 1} \tag{3.8}$$

For reals, we replace $(\alpha-1)!$ with $\Gamma(\alpha)$. Hence

$$p(D) = \frac{[(\alpha_1)\cdots(\alpha_1+N_1-1)]\,[(\alpha_0)\cdots(\alpha_0+N_0-1)]}{(\alpha)\cdots(\alpha+N-1)} \tag{3.9}$$

$$= \frac{\Gamma(\alpha)}{\Gamma(\alpha+N)} \cdot \frac{\Gamma(\alpha_1+N_1)}{\Gamma(\alpha_1)} \cdot \frac{\Gamma(\alpha_0+N_0)}{\Gamma(\alpha_0)} \tag{3.10}$$

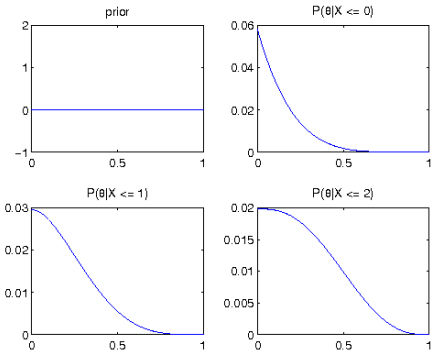


Figure 3.1: Prior and posterior for censored data problem

3.1.3 Posterior predictive for Beta-Binomial model

We have

$$
\begin{aligned}
p(X=1|D) &= Bb(1|\alpha_1', \alpha_0', 1) && (3.11)\\
&= \frac{\Gamma(\alpha_1'+\alpha_0')}{\Gamma(\alpha_1')\Gamma(\alpha_0')}\frac{\Gamma(\alpha_1'+1)\Gamma(\alpha_0'+1-1)}{\Gamma(\alpha_1'+\alpha_0'+1)}\binom{1}{1} && (3.12)\\
&= \frac{\Gamma(\alpha_1'+\alpha_0')}{\Gamma(\alpha_1')\Gamma(\alpha_0')(\alpha_1'+\alpha_0')\Gamma(\alpha_1'+\alpha_0')}\alpha\Gamma(\alpha_1')\Gamma(\alpha_0') && (3.13)\\
&= \frac{\alpha_1'}{\alpha_1'+\alpha_0'} && (3.14)
\end{aligned}
$$

3.1.4 Beta updating from censored likelihood

We have

$$
\begin{aligned}
p(\theta|X<3) &\propto p(\theta, X<3) = p(\theta)p(X<3|\theta) = p(\theta)[\sum_{j=0}^{2} p(X=j|\theta)] && (3.15)\\
&= Beta(\theta|a,b)\sum_{j=0}^{2} Bin(j|n,\theta) && (3.16)\\
&\propto \sum_{j=0}^{2} Beta(\theta|j+a, n+b-j) && (3.17)
\end{aligned}
$$

See Figure 3.1 for a plot of the prior and the mixture posterior after we incorporate each term. (We have normalized the distribution numerically.) See below for the code.

Listing 3.1: betaCensoredPost.m

```
thetas = 0:0.01:1;
a = 1; b = 1; n = 5;
figure(1);clf;
ps =  betapdf(thetas, a, b);
subplot(2,2,1)
plot(thetas, normalize(ps))
title('prior')
ps = zeros(size(thetas));
for j=0:2
  ps = ps + betapdf(thetas, j+a, n-j + b);
  subplot(2,2,j+2)
  plot(thetas, normalize(ps))
  title(sprintf('P(%s|X <= %d)', '\theta', j))
end
```

3.1.5 Uninformative prior for log-odds ratio

We have

$$
\begin{aligned}
p_\theta(\theta) &= p_\phi(\phi)|\frac{d\phi}{d\theta}| && (3.18)\\
&= 1 \times \frac{1-\theta}{\theta}[\frac{1}{1-\theta} + \frac{\theta}{(1-\theta)^2}] && (3.19)\\
&= \frac{1}{\theta(1-\theta)} && (3.20)\\
&\propto Be(\theta|0,0) && (3.21)
\end{aligned}
$$

3.1.6 MLE for the Poisson distribution

The log-likelihood is as follows (dropping terms independent of λ)

$$
\ell(\lambda) = \sum_i \log(e^{-\lambda}\lambda^{x_i}) = -n\lambda + (\sum_i x_i)\log\lambda \tag{3.22}
$$

Taking derivatives and equating to zero yields

$$
\begin{aligned}
\frac{d}{d\lambda}\ell(\lambda) &= -n + \frac{\sum_i x_i}{\lambda} = 0 && (3.23)\\
\lambda &= \frac{\sum_i x_i}{n} && (3.24)
\end{aligned}
$$

3.1.7 Bayesian analysis of the Poisson distribution

1. The posterior is $\mathrm{Ga}(\lambda|a+\sum_i x_i, b+n)$ as we show below.

$$
p(\lambda|D) \propto \lambda^{a-1}e^{-\lambda b}\left[\prod_i e^{-\lambda}\lambda^{x_i}\right] = \lambda^{a+\sum_i x_i - 1}e^{-\lambda(b+n)} \tag{3.25}
$$

2. The posterior mean is

$$
E[\lambda|D] = \frac{a+\sum_i x_i}{b+n} \tag{3.26}
$$

   so tends to the MLE as $a \to 0$ and $b \to 0$.

3.1.8 MLE for the uniform distribution

1. The MLE is the smallest interval that contains all the data, which is exactly equal to the point that is furthest away from the origin:

$$
\hat{a} = max_{i=1}^n |x_i| \tag{3.27}
$$

2. The predictive density using the plug-in rule is

$$
p(x_{n+1}) = \begin{cases} \frac{1}{2\hat{a}} & \text{if } x_{n+1} \in [-\hat{a}, +\hat{a}] \\ 0 & \text{otherwise} \end{cases} \tag{3.28}
$$

3. The problem with this approach is that it predicts that it is impossible for x_{n+1} to fall outside the range of the training data. A conceptually simple solution is to put a prior on a and to integrate it out when predicting the future:

$$
p(x_{n+1}|x_{1:n}) = \int_0^\infty p(x_{n+1}|a)p(a|x_{1:n})da = \int_{|x_{n+1}|}^\infty \frac{1}{2a}p(a|x_{1:n})da \tag{3.29}
$$

   where the second equality follows since $p(x_{n+1}|a) = 0$ if $a < |x_{n+1}|$. A suitable conjugate prior would be the pareto distribution (see Exercise **??**).

3.1.9 Conjugate analysis of the uniform distribution

The posterior for θ is

$$p(\theta|D) = \frac{p(D,\theta)}{p(D)} \tag{3.30}$$

$$= \begin{cases} \frac{Kb^K}{\theta^{N+K+1}} \frac{(N+K)b^N}{K} = \frac{(N+K)b^{N+K}}{\theta^{N+K+1}} & \text{if } m \le b, \theta \ge m \\ \frac{Kb^K}{\theta^{N+K+1}} \frac{(N+K)m^{N+K}}{Kb^K} = \frac{(N+K)m^{N+K}}{\theta^{N+K+1}} & \text{if } m > b, \theta \ge m \\ 0 & \text{if } \theta < m \end{cases} \tag{3.31}$$

Let $c = max(m, b)$. Then we can write the posterior more succintly as

$$p(\theta|D) = Pa(\theta|c, N+K) \tag{3.32}$$

As we get more samples, we become more sure that θ is not much greater than c. For an interesting application of this, see Exercise **??**.

3.1.10 Taxicab (tramcar) problem

See `paretoDemoTaxicab` for some code.

1. If the prior is $p(\theta) = Pa(\theta|b, K)$, then the posterior is $p(\theta|D) = Pa(\theta|c, N+K)$, where $c = max(m, b)$ and $m = max(D)$. With a non-informative prior $b = K = 0$, this is $Pa(\theta|m, N)$. If $D = \{100\}$, so $m = 100, N = 1$, then the posterior is $Pa(\theta|m, 1)$, i.e.,

$$p(\theta|D) = Pa(\theta|m, 1) = \frac{m}{\theta^2}\delta(\theta \ge m) \tag{3.33}$$

2. The posterior mean is $(Nm)/(N-1)$ which does not exist if $N = 1$. The posterior mode is $m = 100$. The posterior median is $2^{1/N}m = 2m = 200$.

3. The predictive density is just the evidence with the updated hyperparameters.

$$p(x|D) = p(x|c, N+k) \tag{3.34}$$

$$= \begin{cases} \frac{N+K}{(N+K+1)c} & \text{if } x \le c \\ \frac{(N+K)c^{N+K}}{(N+K+1)x^{N+K+1}} & \text{if } x > c \end{cases} \tag{3.35}$$

   For $N = 1$, $K = 0$, $c = m$, we have

$$p(x|D) = \begin{cases} \frac{1}{2m} & \text{if } x \le m \\ \frac{m}{2x^2} & \text{if } x > m \end{cases} \tag{3.36}$$

4. We get $p(x = 100|D) = 1/200 = 0.005$, $p(x = 50|D) = 1/200$, and $p(x = 150|D) = 100/(2 * 150^2) = 0.0022$.

5. We could put a reasonable upper bound on θ based on the size of the city and other covariates.

3.1.11 Bayesian analysis of the exponential distribution

1. The loglikelihood is

$$\ell(\theta) = N \log \theta - \theta \sum_i x_i \tag{3.37}$$

   Optimizing we get

$$\frac{d}{d\theta}\ell(\theta) = \frac{N}{\theta} - \sum_i x_i = 0 \tag{3.38}$$

$$\hat{\theta} = \frac{\sum_{i=1}^N x_i}{N} \tag{3.39}$$

2. $\hat{\theta}_{mle}(\mathcal{D}) = 1/\overline{x} = 1/5$.

3. $\mathbb{E}[\theta] = 1/\lambda = 1/3$ so $\hat{\lambda} = 3$

4. The prior is

$$p(\theta) = \theta^3 e^{-3\theta} \tag{3.40}$$

The likelihood is

$$p(\mathcal{D}|\theta) \propto e^{-\theta x_1} e^{-\theta x_2} e^{-\theta x_2} = e^{-15\theta} \tag{3.41}$$

So the posterior is

$$p(\theta|\mathcal{D}) \propto \theta^3 e^{-3\theta} e^{-15\theta} = \mathrm{Ga}(\theta|4, 18) \tag{3.42}$$

5. Yes, this prior is conjugate, since the exponential is a special case of the Gamma.

6. The posterior mean is

$$\frac{4}{18} = \frac{2}{9} = \frac{1}{3}\frac{3}{3+15} + \frac{1}{5}\frac{15}{3+15} \tag{3.43}$$

7. The posterior mean is a compromise between the prior mean (1/3) and the MLE (1/5). This is a more reasonable guess than the MLE since the sample size is small, so we should rely on our expert prior knowledge (although with such a simple one-parameter prior, we were not able to encode how strongly we trusted this expert).

3.1.12 MAP estimation for the Bernoulli with non-conjugate priors

1. Since the prior assigns non-zero probability only to $\theta = 0.4$ or $\theta = 0.5$, we just need to compare the following two estimates

$$\hat{\theta} = \arg \max_{\theta \in \{0.4, 0.5\}} \theta^{N_1} (1-\theta)^{N_0} 0.5 \tag{3.44}$$

The MAP estimate is 0.5 whenever

$$\begin{aligned}
0.4^{N_1} 0.6^{N_0} &< 0.5^{N_1} 0.5^{N_0} && (3.45)\\
0.4^{N_1/N} 0.6^{1-N_1/N} &< 0.5 && (3.46)\\
\frac{N_1}{N}\log(0.5) + (1 - \frac{N_1}{N})\log 0.6 &< \log 0.5 && (3.47)\\
\log\frac{0.6}{0.5} &< \frac{N_1}{N}\log\frac{0.6}{0.4} && (3.48)
\end{aligned}$$

Hence

$$\hat{\theta} = \begin{cases} 0.4 & \text{if } N_1/N < \log(1.2)/\log(1.5) \\ 0.5 & \text{if } N_1/N > \log(1.2)/\log(1.5) \end{cases} \tag{3.49}$$

Note that the boundary case $N_1/N = \log(1.2)/\log(1.5)$ cannot occur because the RHS is irrational.

2. When N is small, the MAP estimate under the biased-coin prior will be 0.4, which is closer to the truth than under the beta prior, since the biased-coin prior only has to rule out $\theta = 0.5$. However, when N is large, the beta prior is preferred, since the MAP estimate under this prior converges to the MLE, which will converge towards the true value of 0.41; the MAP estimate under the biased-coin prior will always have an error of 0.01, and is therefore not a consistent estimator.

3.1.13 Posterior predictive distribution for the Dirichlet-multinomial model

Let $\beta_k = \alpha_k + N_k^{old}$ be the updated hyperparameters. Plugging in the expression for the marginal likelihood of a Dirichlet yields

$$\begin{aligned}
p(D_{new}|D_{old}, \alpha) &= p(D_{new}|\beta) && (3.50)\\
&= \frac{\Gamma(N_{old} + \sum_{k=1} \alpha_k)}{\Gamma(N_{old} + N_{new} + \sum_k \alpha_k)} \prod_k \frac{\Gamma(N_k^{old} + N_k^{new} + \alpha_k)}{\Gamma(N_k^{old} + \alpha_k)} && (3.51)\\
&= \frac{Z_{dir}(\mathbf{N}^{old} + \mathbf{N}^{new} + \boldsymbol{\alpha})}{Z_{dir}(\mathbf{N}^{old} + \boldsymbol{\alpha})} && (3.52)
\end{aligned}$$

where

$$Z_{dir}(\boldsymbol{\alpha}) = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)} \tag{3.53}$$

Here is a more brute force way to compute the same answer.

$$\begin{aligned} p(D_{new}|D_{old}, \vec{\alpha}) &= \frac{p(D_{new}, D_{old}|\vec{\alpha})}{p(D_{old}|\vec{\alpha})} && (3.54)\\ &= \left[\frac{\Gamma(\sum_{k=1} \alpha_k)}{\Gamma(N_{old} + N_{new} + \sum_k \alpha_k)} \prod_k \frac{\Gamma(N_k^{old} + N_k^{new} + \alpha_k)}{\Gamma(\alpha_k)}\right] && (3.55)\\ &\times \left[\frac{\Gamma(N_{old} + \sum_k \alpha_k)}{\Gamma(\sum_k \alpha_k)} \prod_k \frac{\Gamma(\alpha_k)}{\Gamma(N_k^{old} + \alpha_k)}\right] && (3.56)\\ &= \frac{\Gamma(N_{old} + \sum_{k=1} \alpha_k)}{\Gamma(N_{old} + N_{new} + \sum_k \alpha_k)} \prod_k \frac{\Gamma(N_k^{old} + N_k^{new} + \alpha_k)}{\Gamma(N_k^{old} + \alpha_k)} && (3.57) \end{aligned}$$

3.1.14 Posterior predictive for Dirichlet-multinomial

- We have $p = \frac{260+10}{27\times 10+2000} = 0.1189$.
- We have $p = \frac{87+10}{270+2000}\frac{100+10}{270+2001} = 0.002070$.

3.1.15 Setting the beta hyper-parameters

We solve the two simultaneous equations as follows

$$\begin{aligned} m &= a/(a+b) && (3.58)\\ a &= ma + mb && (3.59)\\ a(1-m) &= mb && (3.60)\\ a &= \frac{m}{1-m} b && (3.61) \end{aligned}$$

and

$$\begin{aligned} v &= \frac{m(1-m)}{a+b+1} && (3.62)\\ (a+b+1)v &= m(1-m) && (3.63)\\ &= (\frac{m}{1-m} b + b + 1)v = \frac{b+1-m}{1-m} v && (3.64)\\ b+1-m &= \frac{m(1-m)^2}{v} && (3.65)\\ b &= \frac{m(1-m)^2}{v} + m - 1 && (3.66)\\ a &= \frac{m}{1-m}\left(\frac{m(1-m)^2}{v} + m - 1\right) && (3.67) \end{aligned}$$

If $m = 0.7$ and $v = 0.2^2$, then $\alpha_1 = 2.975$ and $\alpha_2 = 1.275$.

3.1.16 Setting the beta hyper-parameters II

If $m = \frac{\alpha}{\alpha+\beta}$ then $\beta = \frac{\alpha(1-m)}{m}$. Define

$$I(\alpha) = \int_l^u \text{Beta}(\theta|\alpha, \frac{\alpha(1-m)}{m}) d\theta \tag{3.68}$$

Then we use numerical optimization to solve

$$\alpha^* = \min(0.95 - I(\alpha))^2 \tag{3.69}$$

Below is some code. We find $\alpha = 4.5$ and $\beta = 25.5$, so the effective prior size is 30.

Listing 3.2: Listing of `betaParamsCI`

```
pdf = @(theta,alpha) betapdf(theta, alpha, ((1-m)/m)*alpha);
I = @(alpha) quad(@(theta) pdf(theta,alpha), l, u);
f = @(alpha) (0.95-I(alpha))^2;
opts.numDiff = 1;
opts.verbose = 0;
init = 5;
alpha = minFunc(f, init, opts)
```

3.1.17 Marginal likelihood for beta-binomial under uniform prior

We have

$$
\begin{aligned}
p(\mathcal{D}) &= \binom{N}{N_1} \frac{B(\alpha_1 + N_1, \alpha_2 + N_2)}{B(\alpha_1, \alpha_2)} && (3.70)\\
&= \frac{N!}{N_1! N_2!} \frac{B(1 + N_1, 1 + N_2)}{B(1,1)} && (3.71)\\
&= \frac{\Gamma(N+1)}{\Gamma(N_1+1)\Gamma(N_2+1)} \frac{\Gamma(1+N_1)\Gamma(1+N_2)}{\Gamma(N+2)} \frac{\Gamma(2)}{\Gamma(1)\Gamma(1)} && (3.72)\\
&= \frac{\Gamma(N+1)}{\Gamma(N+2)} = \frac{\Gamma(N+1)}{(N+1)\Gamma(N+1)} = \frac{1}{N+1} && (3.73)
\end{aligned}
$$

where we used the facts that $N = N_1 + N_2$, $\Gamma(1) = \Gamma(2) = 1$, $\Gamma(N+1) = N!$, and $\Gamma(1 + (1 + N)) = (N+1)\Gamma(N+1)$.

3.1.18 Bayes factor for coin tossing

We have

$$
p(N_1 = 9|N = 10, M_0) = \mathrm{Bin}(9|10, 0.5) = \binom{10}{9} 0.5^{10} = 0.0098 \tag{3.74}
$$

Now considrer the alternative hypothesis. If $N_1 \sim \mathrm{Bin}(N, \theta)$ and $\theta \sim \mathrm{Beta}(1,1)$, one can show (Exercise **??**) that the marginal likelihood is $p(N_1|N) = 1/(N+1)$. Hence

$$
p(N_1 = 9|N = 10, M_1) = \binom{N}{N_1} \int \theta^{N_1} (1-\theta)^{N-N_1} d\theta = \frac{1}{N+1} = 0.0909 \tag{3.75}
$$

So $BF_{1,0} = 0.0909/0.0098 = 9.3$, which is moderate evidence in favor of the biased coin hypothesis. If we toss the coin $N = 100$ times and observe $N_1 = 90$ heads, we find $BF_{10} = 10^{15}$, which is decisive evidence for bias (see `coinBayesFactorDemo` for the code).

3.1.19 Irrelevant features with naive Bayes

1. By Bayes rule, we have

$$
\begin{aligned}
\frac{p(c=1|\mathbf{x})}{p(c=2|\mathbf{x})} &= \frac{p(\mathbf{x}|c=1)p(c=1)}{p(\mathbf{x}|c=2)p(c=2)} = \frac{p(\mathbf{x}|c=1)}{p(\mathbf{x}|c=2)} && (3.76)\\
& && (3.77)
\end{aligned}
$$

   which is just the likelihood ratio. Hence

$$
\log \frac{p(c=1|\mathbf{x})}{p(c=2|\mathbf{x})} = \boldsymbol{\phi}(\mathbf{x}_i)^T (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2) \tag{3.78}
$$

2. If $\beta_{1,w} = \beta_{2,w}$, then $\beta_{1,w} - \beta_{2,w} = 0$, so word w will be ignored. This is equivalent to requiring the log odds ratio, $\log \frac{\theta_{cw}}{1-\theta_{cw}}$, to be the same for both classes.

3. The estimates will be

$$
\hat{\theta}_{1w} = \frac{1+n_1}{2+n_1}, \;\; \hat{\theta}_{2w} = \frac{1+n_2}{2+n_2} \tag{3.79}
$$

   We see that these are different, since $n_1 \neq n_2$. Hence $\frac{\theta_{1w}}{1-\theta_{1w}} \neq \frac{\theta_{2w}}{1-\theta_{2w}}$, so the word will not be ignored. If we use the MLE, we get $n_1/n_1 = n_2/n_2 = 1$, so this problem does not occur. However, the MLE results in overfitting.

4. We can use feature selection methods to try to remove irrelevant words. This is similar to imposing the prior that says $\theta_{c,w}$ is the same across classes.

3.1.20 Class conditional densities for binary data

1. We can represent $p(\mathbf{x}|y=c)$ as a vector with $2^d - 1$ elements; we need to specify the probability of each particular bit configuration $\mathbf{x}$. For example, if $d = 3$, we need C histograms (one per class) which store the probabilities of seeing $\mathbf{x} = (0,0,0), \ldots, (1,1,1)$. Hence the total number of parameters in the model is $C(2^d - 1)$ (-1 because of the sum-to-one constraint). We cannot use a multivariate Gaussian since the data is binary.

2. If the training set size is low, the NB model is likely to work better, since it has fewer parameters, so is less likely to overfit. Note that many people said they would prefer the complex model for small N, because it is more "accurate". But this is wrong. If we used cross validation or Bayesian model selection, we would not pick the full model given a small sample size.

3. If the sample size is large, the full model is likely to work better, since it is a more accurate model, and we have enough data to reliably estimate all the parameters without overfitting.

4. Fitting both models takes $O(ND)$ time: you just iterate over the training cases, and update the counts. We give more details below.

   Consider fitting the NB model. We have $\hat{\theta}_{jc} = \frac{N_{jc}}{N_c}$, where N_{jc} is the number of times bit j turns on in class c, and $j = 1 : D$. This can be computed in $O(ND)$ time as follows:

   ```
   for i=1:N
     let c=y(i)
     for j=1:D
        if X(i,j)==1, then N(j,c)++
   ```

   Now consider fitting the full model. We have $\hat{\theta}_{kc} = \frac{N_{kc}}{N_c}$, where N_{kc} is the number of times bitvector k appears in class c, for $k = 1 : 2^D$. This can be computed in $O(ND)$ time as follows:

   ```
   for i=1:N
     let c=y(i)
     let k=X(i,:)  // this takes O(D) time
     N(k,c)++
   ```

   The line `k=X(i,:)` converts the bit vector $\mathbf{x}_i$ to an integer index, and is assumed to take $O(D)$ time. The key point is that, although we have $O(2^D)$ parameters, most of them will be 0, so they do not need to be updated. Of course, if $N \gg 2^D$, then all the parameters will be non-zero, so it will take $O(2^D)$ work to fit the model; but $O(ND)$ is still an accurate description of the complexity.

5. Both models take $O(D)$ time to evaluate $p(\mathbf{x}|y)$. For NB we have

$$p(y=c|\mathbf{x}, \boldsymbol{\theta}) = \prod_{j=1}^{D} \theta_{jc}^{\mathbb{I}(x_j=1)} \tag{3.80}$$

   For the full model we have

$$p(y=c|\mathbf{x}, \boldsymbol{\theta}) = \theta_{k(\mathbf{x}),c} \tag{3.81}$$

   where $k(\mathbf{x})$ is the integer encoding of the bit vector $\mathbf{x}$.

6. We can always handle missing data in a generative classifier by marginalizing out the missing features. We need to compute $p(y|\mathbf{x}_v) \propto p(\mathbf{x}_v|y=c)p(y=c)$, where $p(\mathbf{x}_v|y=c) = \sum_{\mathbf{x}_h} p(\mathbf{x}_v, \mathbf{x}_h|y=c)$. Computing the latter takes $O(v)$ time in a NB classifier, since the features are conditionally independent, so we can just drop the hidden ones. But it takes $O(2^h)$ time to compute this in the full model, since we must marginalize over all 2^h configurations. So the NB model is faster at handling missing data.

3.1.21 Mutual information for naive Bayes classifiers with binary features

$$
\begin{aligned}
I(X_j, Y) &= \sum_{x=0}^{1}\sum_{c=1}^{C} p(X_j = x, y = c) \log \frac{p(X_j = x|y = c)p(y = c)}{p(X_j = x)p(y = c)} && (3.82)\\
&= \sum_{x=0}^{1}\sum_{c} p(X_j = x|y = c)p(y = c) \log \frac{p(X_j = x|y = c)}{p(X_j = x)} && (3.83)\\
&= \sum_{c} p(X_j = 1|y = c)p(y = c) \log \frac{p(X_j = 1|y = c)}{p(X_j = 1)} && (3.84)\\
&\quad + \sum_{c} p(X_j = 0|y = c)p(y = c) \log \frac{p(X_j = 0|y = c)}{p(X_j = 0)} && (3.85)\\
&= \sum_{c} \left[\theta_{jc}\pi_c \log \frac{\theta_{jc}}{\theta_j} + (1 - \theta_{jc})\pi_c \log \frac{1 - \theta_{jc}}{1 - \theta_j}\right] && (3.86)
\end{aligned}
$$

3.1.22 Fitting a naive bayes spam filter by hand

$\theta_{\text{spam}} = 3/7$, $\theta_{\text{secret—spam}} = 2/3$, $\theta_{\text{secret—non-spam}} = 1/4$, $\theta_{\text{sports—non-spam}} = 2/4$, $\theta_{\text{dollar—spam}} = 1/3$.

Chapter 4

Gaussian models

4.1 Solutions

4.1.1 Uncorrelated does not imply independent

We have

$$
\begin{aligned}
E[X] &= 0 && (4.1)\\
\text{var}\,[X] &= (1--1)^2/12 = 1/3 && (4.2)\\
E[Y] &= E[X^2] = \text{var}\,[X] + (E[X])^2 = 1/3 + 0 && (4.3)\\
E[XY] &= \int p(x)x^3 dx = \frac{1}{2}\left[-(\int_{-1}^{0} x^3 dx) + (\int_{0}^{1} x^3 dx)\right] && (4.4)\\
&= \frac{1}{2}\left[\frac{-1}{4}[x^4]_{-1}^{0} + \frac{1}{4}[x^4]_{0}^{1}\right] = \frac{1}{8}\left[-1+1\right] = 0 && (4.5)
\end{aligned}
$$

where we have split the integral into two pieces, since x^3 changes sign in the interval $[-1, 1]$. Hence

$$
\rho = \frac{\text{cov}\,[X,Y]}{\sigma_X \sigma_Y}\frac{E[XY] - E[X]E[Y]}{\sigma_X \sigma_Y} = \frac{0}{\sigma_X \sigma_Y} = 0 \quad (4.6)
$$

4.1.2 Uncorrelated does not imply independent unless jointly Gaussian

1.

2. For the mean, we have

$$
\mathbb{E}\left[Y\right] = \mathbb{E}\left[WX\right] = \mathbb{E}\left[X\right]\mathbb{E}\left[X\right] = 0 \quad (4.7)
$$

   For the variance, we have

$$
\begin{aligned}
\text{var}\,[Y] &= \mathbb{E}\left[\text{var}\,[Y|W]\right] + \text{var}\left[\mathbb{E}\left[Y|W\right]\right] && (4.8)\\
&= \mathbb{E}\left[W[\text{var}\,[X]]W\right] + \text{var}\left[W\mathbb{E}\left[X\right]\right] && (4.9)\\
&= \mathbb{E}\left[W^2\right] + 0 = 1 && (4.10)
\end{aligned}
$$

   To show it's Gaussian, we note that Y is a linear combination of Gaussian rv's.

3. To show that $\text{cov}\,[X,Y] = 0$, we use the rule of iterated expectation. First we have

$$
\begin{aligned}
\mathbb{E}\left[XY\right] &= \mathbb{E}\left[\mathbb{E}\left[XY|W\right]\right] && (4.11)\\
&= \sum_{w\in\{-1,1\}} p(w)\mathbb{E}\left[XY|w\right] && (4.12)\\
&= -1\cdot 0.5\cdot\mathbb{E}\left[X\cdot -X\right] + 1\cdot 0.5\cdot\mathbb{E}\left[X\cdot X\right] && (4.13)\\
&= = 0 && (4.14)
\end{aligned}
$$

Then we have

$$\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}[\mathbb{E}[Y|W]] && (4.15)\\
&= \sum_{w\in\{-1,1\}} p(w)\mathbb{E}[Y|w] && (4.16)\\
&= 0.5\cdot\mathbb{E}[-X] + 0.5\cdot\mathbb{E}[X] && (4.17)\\
&= 0 && (4.18)
\end{aligned}$$

Hence

$$\mathrm{cov}[X,Y] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = 0 \tag{4.19}$$

So X and Y are uncorrelated even though they are dependent.

4.1.3 Correlation coefficient is between -1 and +1

We have

$$\begin{aligned}
0 &\leq \mathrm{var}\left[\frac{X}{\sigma_X} + \frac{Y}{\sigma_Y}\right] && (4.20)\\
&= \mathrm{var}\left[\frac{X}{\sigma_X}\right] + \mathrm{var}\left[\frac{Y}{\sigma_Y}\right] + 2\mathrm{cov}\left[\frac{X}{\sigma_X}, \frac{Y}{\sigma_Y}\right] && (4.21)\\
&= \frac{\mathrm{var}[X]}{\sigma_X^2} + \frac{\mathrm{var}[Y]}{\sigma_Y^2} + 2\mathrm{cov}\left[\frac{X}{\sigma_X}, \frac{Y}{\sigma_Y}\right] && (4.22)\\
&= 1 + 1 + 2\rho = 2(1+\rho) && (4.23)
\end{aligned}$$

Hence $\rho \geq -1$. Similarly,

$$0 \leq \mathrm{var}\left[\frac{X}{\sigma_X} - \frac{Y}{\sigma_Y}\right] = 2(1-\rho) \tag{4.24}$$

so $\rho \leq 1$.

4.1.4 Correlation coefficient for linearly related variables is ± 1

Let $\mathbb{E}[X] = \mu$ and $\mathrm{var}[X] = \sigma^2$, and assume $a > 0$. Then we have

$$\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}[aX+b] = a\mathbb{E}[X] + b = a\mu + b && (4.25)\\
\mathrm{var}[Y] &= \mathrm{var}[aX+b] = a^2\mathrm{var}[X] = a^2\sigma^2 && (4.26)\\
\mathbb{E}[XY] &= \mathbb{E}[X(aX+b)] = \mathbb{E}\left[aX^2 + bX\right] = a\mathbb{E}\left[X^2\right] + b\mathbb{E}[X] = a(\mu^2+\sigma^2) + b\mu && (4.27)\\
\mathrm{cov}[X,Y] &= \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = a(\mu^2+\sigma^2) + b\mu - \mu(a\mu+b) = a\sigma^2 && (4.28)\\
\rho &= \frac{\mathrm{cov}[X,Y]}{\sqrt{\mathrm{var}[X]\,\mathrm{var}[Y]}} = \frac{a\sigma^2}{\sqrt{\sigma^2}\sqrt{a^2\sigma^2}} = 1 && (4.29)
\end{aligned}$$

If $Y = aX + b$, where $a < 0$, we find $\rho = -1$, since $a/\sqrt{a^2} = -a/|a| = -1$.

4.1.5 Normalization constant for a multidimensional Gaussian

Let $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, so

$$\boldsymbol{\Sigma}^{-1} = \mathbf{U}^{-T}\boldsymbol{\Lambda}^{-1}\mathbf{U}^{-1} = \mathbf{U}\boldsymbol{\Lambda}^{-1}\mathbf{U} = \sum_{i=1}^{p}\frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T \tag{4.30}$$

Hence

$$\begin{aligned}
(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) &= (\mathbf{x}-\boldsymbol{\mu})^T\left(\sum_{i=1}^{p}\frac{1}{\lambda_i}\mathbf{u}_i\mathbf{u}_i^T\right)(\mathbf{x}-\boldsymbol{\mu}) && (4.31)\\
&= \sum_{i=1}^{p}\frac{1}{\lambda_i}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{u}_i\mathbf{u}_i^T(\mathbf{x}-\boldsymbol{\mu}) = \sum_{i=1}^{p}\frac{y_i^2}{\lambda_i} && (4.32)
\end{aligned}$$

where $y_i \triangleq \mathbf{u}_i^T(\mathbf{x} - \boldsymbol{\mu})$. The $\mathbf{y}$ variables define a new coordinate system that is shifted (by $\boldsymbol{\mu}$) and rotated (by $\mathbf{U}$) with respect to the original x coordinates: $\mathbf{y} = \mathbf{U}(\mathbf{x} - \boldsymbol{\mu})$. Hence $\mathbf{x} = \mathbf{U}^T\mathbf{y} + \boldsymbol{\mu}$.

The Jacobian of this transformation, from $\mathbf{y}$ to $\mathbf{x}$, is a matrix with elements

$$J_{ij} = \frac{\partial x_i}{\partial y_j} = U_{ji} \tag{4.33}$$

so $\mathbf{J} = \mathbf{U}^T$ and $|\mathbf{J}| = 1$.

So

$$\int \exp(-\frac{1}{2}(\mathbf{x} - \mu)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}))d\mathbf{x} = \int \prod_i \exp(-\frac{1}{2}\sum_i \frac{y_i^2}{\lambda_i})dy_i|\mathbf{J}| \tag{4.34}$$

$$= \prod_i \sqrt{2\pi\lambda_i} = |2\pi\boldsymbol{\Sigma}| \tag{4.35}$$

4.1.6 Bivariate Gaussians

We have

$$|\Sigma| = \sigma_1^2\sigma_2^2 - \rho^2\sigma_1^2\sigma_2^2 = \sigma_1^2\sigma_2^2(1 - \rho^2) \tag{4.36}$$

$$|\Sigma|^{\frac{1}{2}} = \sigma_2\sigma_2\sqrt{1 - \rho^2} \tag{4.37}$$

$$\Sigma^{-1} = \frac{1}{\sigma_1^2\sigma_2^2(1 - \rho^2)}\begin{pmatrix} \sigma_1^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{4.38}$$

so $(x - \mu)^T\Sigma^{-1}(x - \mu)$ expands to

$$\begin{pmatrix} x_1 - \mu_1 & x_2 - \mu_2 \end{pmatrix}\begin{pmatrix} \frac{\sigma_2^2}{\sigma_1^2\sigma_2^2(1-\rho^2)} & -\frac{\rho\sigma_1\sigma_2}{\sigma_1^2\sigma_2^2(1-\rho^2)} \\ -\frac{\rho\sigma_1\sigma_2}{\sigma_1^2\sigma_2^2(1-\rho^2)} & \frac{\sigma_1^2}{\sigma_1^2\sigma_2^2(1-\rho^2)} \end{pmatrix}\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix} \tag{4.39}$$

$$= \frac{1}{1 - \rho^2}\left(\frac{(x_1 - \mu_1)^2}{\sigma_1^2} + \frac{(x_2 - \mu_2)^2}{\sigma_2^2} - 2\rho\frac{(x_1 - \mu_1)}{\sigma_1}\frac{(x_2 - \mu_2)}{\sigma_2}\right) \tag{4.40}$$

4.1.7 Conditioning a bivariate Gaussian

1. We have that $P(X_2|x_1) = \mathcal{N}(X_2|\mu_{2|1}, \Sigma_{2|1})$, where

$$\mu_{2|1} = \mu_2 + \frac{\sigma_{12}}{\sigma_1^2}(x_1 - \mu_1) = \mu_2 + \frac{\sigma_2\rho}{\sigma_1}(x_1 - \mu_1)$$

$$\sigma_{2|1}^2 = \sigma_2^2 - \frac{\sigma_{12}^2}{\sigma_2^2} = \sigma_2^2(1 - \rho^2)$$

2. Now $\sigma_{12} = \rho$, so the above becomes

$$\mu_{2|1} = \mu_2 + \rho(x_1 - \mu_1)$$

$$\sigma_{2|1}^2 = 1 - \rho^2$$

4.1.8 Whitening vs standardizing

Code is below.

Listing 4.1: heightWeightWhiten.m

```
rawdata = dlmread('heightWeightData.txt'); % comma delimited file
data.Y = rawdata(:,1); % 1=male, 2=female
data.X = [rawdata(:,2) rawdata(:,3)]; % height, weight

maleNdx = find(data.Y == 1);

X = data.X(maleNdx,:);
XS = standardize(X);

% Whiten
Sigma = cov(X);
```

```
mu = mean(X);
n = size(X,1);
[U,D] = eig(Sigma);
A = sqrt(inv(D))*U';
XW = X'; % each column is a case
XW = A*(XW-repmat(mu(:),1,n));
XW = XW'; % reach row is a case

% Plot data
XX = {X, XS, XW};
ttl = {'raw', 'standarized', 'whitened'};
figure
for j=1:length(XX)
  X = XX{j};
  % plot identity of each male
  subplot(1,3,j)
  N = size(X,1);
  for i=1:N
    plot(X(i,1), X(i,2));
    hold on
    str = sprintf('%d', i);
    text(X(i,1), X(i,2), str);
  end
  hold on
  mu = mean(X); Sigma = cov(X);
  gaussPlot2d(mu, Sigma);
  if j>2, axis equal, end
  title(ttl{j});
end
```

4.1.9 Sensor fusion with known variances in 1d

We just modify Section **??** to handle different variances. Define the sufficient statistics as

$$\overline{y}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1} y_i^{(1)}, \;\; \overline{y}_2 = \frac{1}{n_2}\sum_{i=1}^{n_2} y_i^{(2)}, \tag{4.41}$$

Define the prior as

$$\mu_\mu = 0, \Sigma_\mu = \infty \tag{4.42}$$

Define the likelihood as

$$\mathbf{A} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \boldsymbol{\mu}_y = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{\Sigma}_y = \begin{pmatrix} \frac{v_1}{n_1} & 0 \\ 0 & \frac{v_2}{n_2} \end{pmatrix} \tag{4.43}$$

Now we just apply the equations. The posterior precision is a sum of the precisions of each sensor:

$$\Sigma_{\mu|y}^{-1} = \mathbf{A}^T \boldsymbol{\Sigma}_y^{-1} \mathbf{A} = \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} \frac{v_1}{n_1} & 0 \\ 0 & \frac{v_2}{n_2} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{n_1}{v_1} + \frac{n_2}{v_2} \tag{4.44}$$

The posterior mean is a weighted sum of the observed values from each sensor:

$$\mu_{\mu|y} = \Sigma_{\mu|y}^{-1} \left(\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} \frac{v_1}{n_1} & 0 \\ 0 & \frac{v_2}{n_2} \end{pmatrix} \begin{pmatrix} \overline{y}_1 \\ \overline{y}_2 \end{pmatrix} \right) = \Sigma_{\mu|y}^{-1} \left(\frac{n_1 \overline{y}_1}{v_1} + \frac{n_2 \overline{y}_2}{v_2} \right) \tag{4.45}$$

4.1.10 Derivation of information form formulae for marginalizing and conditioning

We will use the following fact. The $\boldsymbol{\eta}$ parameter of the joint is given by

$$\begin{pmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \tag{4.46}$$

so

$$\boldsymbol{\eta}_1 = \boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 + \boldsymbol{\Lambda}_{12}\boldsymbol{\mu}_2 \tag{4.47}$$

$$\boldsymbol{\eta}_2 = \boldsymbol{\Lambda}_{21}\boldsymbol{\mu}_1 + \boldsymbol{\Lambda}_{22}\boldsymbol{\mu}_2 \tag{4.48}$$

We have

$$\begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}^{-1} \tag{4.49}$$

$$= \begin{pmatrix} (\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22})^{-1} & -(\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22})^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} \\ -\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}(\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22})^{-1} & \boldsymbol{\Sigma}_{22}^{-1} + \boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}(\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22})^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-2} \end{pmatrix} \tag{4.50}$$

where the top left is

$$\boldsymbol{\Lambda}_{11} = (\boldsymbol{\Sigma}/\boldsymbol{\Sigma}_{22})^{-1} = (\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1} \tag{4.51}$$

Hence from Equation **??**

$$\boldsymbol{\Lambda}_{1|2} = \boldsymbol{\Sigma}_{1|2}^{-1} = (\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1} = \boldsymbol{\Lambda}_{11} \tag{4.52}$$

From Equation **??** we have

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \tag{4.53}$$

but from Equation 4.50 and 4.51 we have

$$\boldsymbol{\Lambda}_{12} = -\boldsymbol{\Lambda}_{11}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} \tag{4.54}$$

so

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \tag{4.55}$$

$$\boldsymbol{\eta}_{1|2} = \boldsymbol{\Lambda}_{1|2}\boldsymbol{\mu}_{1|2} = \boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 - \boldsymbol{\Lambda}_{12}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \tag{4.56}$$

$$= \boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 + \boldsymbol{\Lambda}_{12}\boldsymbol{\mu}_2 - \boldsymbol{\Lambda}_{12}\mathbf{x}_2 = \boldsymbol{\eta}_1 - \boldsymbol{\Lambda}_{12}\mathbf{x}_2 \tag{4.57}$$

We will now derive the results for marginalizing in information form. Let

$$\begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix}^{-1} \tag{4.58}$$

$$= \begin{pmatrix} \boldsymbol{\Lambda}_{11}^{-1} + \boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\boldsymbol{\Lambda}/\boldsymbol{\Lambda}_{11})^{-1}\boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1} & -\boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}(\boldsymbol{\Lambda}/\boldsymbol{\Lambda}_{11})^{-1} \\ -(\boldsymbol{\Lambda}/\boldsymbol{\Lambda}_{11})^{-1}\boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1} & (\boldsymbol{\Lambda}/\boldsymbol{\Lambda}_{11})^{-1} \end{pmatrix} \tag{4.59}$$

Hence

$$\boldsymbol{\Lambda}_{22}^m = \boldsymbol{\Sigma}_{22}^{-1} = \boldsymbol{\Lambda}/\boldsymbol{\Lambda}_{11} = \boldsymbol{\Lambda}_{22} - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12} \tag{4.60}$$

$$\boldsymbol{\eta}_2^m = \boldsymbol{\Lambda}_{22}^m\boldsymbol{\mu}_2^m = (\boldsymbol{\Lambda}_{22} - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12})\boldsymbol{\mu}_2 \tag{4.61}$$

$$= \boldsymbol{\Lambda}_{22}\boldsymbol{\mu}_2 - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1}\boldsymbol{\Lambda}_{12}\boldsymbol{\mu}_2 \tag{4.62}$$

$$= (\boldsymbol{\Lambda}_{21}\boldsymbol{\mu}_1 + \boldsymbol{\Lambda}_{22}\boldsymbol{\mu}_2) - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}^{-1}(\boldsymbol{\Lambda}_{11}\boldsymbol{\mu}_1 + \boldsymbol{\Lambda}_{12}\boldsymbol{\mu}_2) \tag{4.63}$$

$$= \boldsymbol{\eta}_2 - \boldsymbol{\Lambda}_{21}\boldsymbol{\Lambda}_{11}\boldsymbol{\eta}_1 \tag{4.64}$$

4.1.11 Derivation of the NIW posterior

Multiplying prior and likelihood we get

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) \propto |\boldsymbol{\Sigma}|^{-\frac{\nu_0+D+2+N}{2}} \mathrm{etr}\left[-\frac{1}{2}\boldsymbol{\Sigma}^{-1}\left(\mathbf{S}_{\overline{x}} + \mathbf{S}_0 + \right.\right. \tag{4.65}$$

$$\left.\left. + N(\boldsymbol{\mu} - \overline{\mathbf{x}})(\boldsymbol{\mu} - \overline{\mathbf{x}})^T + \kappa_0(\boldsymbol{\mu} - \mathbf{m}_0)(\boldsymbol{\mu} - \mathbf{m}_0)^T\right)\right] \tag{4.66}$$

where $\mathrm{etr}(\mathbf{M}) = \exp(\mathrm{tr}(\mathbf{M}))$. Using the hint this becomes

$$p(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathcal{D}) \propto |\boldsymbol{\Sigma}|^{-\frac{\nu_N+D+2}{2}} \mathrm{etr}\left[-\frac{1}{2}\boldsymbol{\Sigma}^{-1}\left(\mathbf{S}_{\overline{x}} + \mathbf{S}_0 + \frac{\kappa_0 N}{\kappa_N}(\overline{\mathbf{x}} - \mathbf{m}_0)(\overline{\mathbf{x}} - \mathbf{m}_0)^T\right)\right] \tag{4.67}$$

$$\times \exp\left[-\frac{\kappa_N}{2}(\boldsymbol{\mu} - \mathbf{m}_N)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{m}_N)\right] \tag{4.68}$$

For the NWI case,

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\mathcal{D}) = \mathrm{NWI}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\mathbf{m}_n, \kappa_n, \nu_n, \mathbf{W}_n) \tag{4.69}$$

$$\mathbf{m}_n = \frac{\kappa_0\mathbf{m}_0 + N\overline{\mathbf{x}}}{\kappa_n} \tag{4.70}$$

$$\kappa_n = \kappa_0 + N \tag{4.71}$$

$$\nu_n = \nu_0 + N \tag{4.72}$$

$$\mathbf{W}_n^{-1} = \mathbf{W}_0^{-1} + \mathbf{S}_{\overline{x}} + \frac{\kappa_0 N}{\kappa_0 + N}(\overline{\mathbf{x}} - \mathbf{m}_0)(\overline{\mathbf{x}} - \mathbf{m}_0)^T \tag{4.73}$$

4.1.12 BIC for Gaussians

1. Using the fact that $\hat{\mathbf{\Sigma}}_{ML} = \mathbf{S}$, we have

$$\begin{aligned} \log p(\mathcal{D}|\hat{\theta}_{ML}) &= -\frac{N}{2}\mathrm{tr}\left(\hat{\mathbf{\Sigma}}^{-1}\mathbf{S}\right) - \frac{N}{2}\log(|\hat{\mathbf{\Sigma}}|) && (4.74)\\ &= -\frac{N}{2}\mathrm{tr}(\mathbf{I}_D) - \frac{N}{2}\log(|\hat{\mathbf{\Sigma}}|) && (4.75)\\ &= -\frac{N}{2}\left(D + \log|\hat{\mathbf{\Sigma}}|\right) && (4.76)\end{aligned}$$

   There are D free parameters for the mean and $D(D+1)/2$ for the covariance, so the BIC score is

$$BIC = -\frac{N}{2}\left(D + \log|\hat{\Sigma}|\right) - \frac{d}{2}\log N \quad (4.77)$$

   where $d = D + D(D+1)/2$.

2. Let $\mathbf{S}_{diag} = \mathrm{diag}(\mathrm{diag}(\mathbf{S})) = \hat{\mathbf{\Sigma}}_{diag}$. Then

$$\mathrm{tr}(\hat{\mathbf{\Sigma}}_{diag}^{-1}\mathbf{S}) = \mathrm{tr}(\hat{\mathbf{\Sigma}}_{diag}^{-1}\mathbf{S}_{diag}) = \mathrm{tr}(\mathbf{I}_D) = D$$

   because the off-diagonal elements of the matrix product get ignored. Hence

$$BIC = -\frac{N}{2}\left(D + \log|\hat{\mathbf{\Sigma}}|\right) - \frac{d}{2}\log N \quad (4.78)$$

   where $d = D + D$ parameters (for the mean and diagonal).

4.1.13 Gaussian posterior credible interval

We want an interval that satisifes

$$p(\ell \leq \mu_n \leq u|D) \geq 0.95 \quad (4.79)$$

where

$$\begin{aligned} \ell &= \mu_n + \Phi^{-1}(0.025)\sigma_n = \mu_n - 1.96\sigma_n && (4.80)\\ u &= \mu_n + \Phi^{-1}(0.9755)\sigma_n = \mu_n + 1.96\sigma_n && (4.81)\end{aligned}$$

where Φ is the cumulative distribution function for the standard normal $\mathcal{N}(0,1)$ distribution, and $\Phi^{-1}(0.025)$ is the value below which 2.5% of the probability mass lies (in Matlab, `norminv(0.025)=-1.96`). and $\Phi^{-1}(0.975)$ is the value below which 97.5% of the probability mass lies (in Matlab, `norminv(0.975)=1.96`). We want to find n such that

$$u - \ell = 1 \quad (4.82)$$

Hence we solve

$$\begin{aligned} 2(1.96)\sigma_n &= 1 && (4.83)\\ \sigma_n^2 &= \frac{1}{4(1.96)^2} && (4.84)\end{aligned}$$

where

$$\sigma_n^2 = \frac{\sigma^2\sigma_0^2}{n\sigma_0^2 + \sigma^2} \quad (4.85)$$

Hence

$$\begin{aligned} n\sigma_0^2 + \sigma^2 &= (\sigma^2\sigma_0^2)4(1.96)^2 && (4.86)\\ n &= \frac{\sigma^2(\sigma_0^2 4(1.96)^2 - 1)}{\sigma_0^2} && (4.87)\\ &= \frac{4(9\times(1.96)^2 - 1)}{9} = 61.0212 && (4.88)\end{aligned}$$

Hence we need at least $n \geq 62$ samples.

4.1.14 MAP estimation for 1D Gaussians

1. Since the mean and mode of a Gaussian are the same, we can just write down the MAP estimate using the standard equations for the posterior mean of a Gaussian, which is

$$E[\mu|D] = \left(\frac{ns^2}{ns^2+\sigma^2}\right)\overline{x} + \left(\frac{\sigma^2}{ns^2+\sigma^2}\right)m \tag{4.89}$$

However, we can also derive the MAP estimate explicitly, as follows. The prior distribution over the mean is

$$p(\mu) = (2\pi s^2)^{-1/2}\exp[-\frac{(\mu-m)^2}{2s^2}] \tag{4.90}$$

Since the samples x_i are taken to be independent, we have:

$$p(D|\theta) = e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2} \tag{4.91}$$

And combining them:

$$\log p(\mu)p(D|\mu) = -\frac{1}{2}\log(2\pi s^2) - \frac{(\mu-m)^2}{2s^2} - \frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2 \tag{4.92}$$

Taking the derivative with respect to μ :

$$\frac{\partial \log(p(\mu)p(D|\mu))}{\partial\mu} = 0 - \frac{2(\mu-m)(1)}{2s^2} - 0 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(2)(x_i-\mu)(-1) \tag{4.93}$$

$$= -\frac{\mu-m}{s^2} + \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu) \tag{4.94}$$

$$= -\frac{\mu}{s^2} + \frac{m}{s^2} + \frac{1}{\sigma^2}\sum_{i=1}^{n}x_i - \frac{n\mu}{\sigma^2} \tag{4.95}$$

Setting this derivative to zero in order to find the maximum:

$$0 = -\frac{\hat{\mu}_{MAP}}{s^2} + \frac{m}{s^2} + \frac{1}{\sigma^2}\sum_{i=1}^{n}x_i - \frac{n\hat{\mu}_{MAP}}{\sigma^2} \tag{4.96}$$

$$\hat{\mu}_{MAP}(\frac{n}{\sigma^2}+\frac{1}{s^2}) = \frac{1}{\sigma^2}\sum_{i=1}^{n}x_i + \frac{m}{s^2} \tag{4.97}$$

$$\hat{\mu}_{MAP} = \frac{\frac{1}{\sigma^2}\sum_{i=1}^{n}x_i + \frac{m}{s^2}}{\frac{n}{\sigma^2}+\frac{1}{s^2}} = \frac{s^2\sum_{i=1}^{n}x_i + m\sigma^2}{ns^2+\sigma^2} \tag{4.98}$$

$$= \left(\frac{ns^2}{ns^2+\sigma^2}\right)\frac{\sum_{i=1}^{n}x_i}{n} + \left(\frac{\sigma^2}{ns^2+\sigma^2}\right)m \tag{4.99}$$

Note: you did not need to derive all this, it was sufficient to look up the equation for the posterior mean/mode in the book.

2. Notice that as n increases, while s, m and σ remain constant, we have $\frac{\sigma^2}{ns^2+\sigma^2} \to 0$ while $\frac{ns^2}{ns^2+\sigma^2} \to 1$, yielding $\hat{\mu}_{MAP} \to \frac{\sum_i x_i}{n} = \hat{\mu}_{MLE}$. That is, when there are many samples, the prior knowledge becomes less and less relevant, and all MAP estimators (for any prior) converge to the maximum likelihood estimator $\hat{\mu}_{MLE}$. This is not surprising: as we have more data, it outweighs our prior speculations.

3. As the prior variance s^2 increases, even for a small number of samples n, we have $\frac{\sigma^2}{ns^2+\sigma^2} \to 0$ while $\frac{ns^2}{ns^2+\sigma^2} \to 1$, again yielding $\hat{\mu}_{MAP} \to \frac{\sum_i x_i}{n} = \hat{\mu}_{MLE}$. That is, when we have an uninformative, almost uniform, prior, we are left only with our data to base out estimation on.

4. On the other hand, if the prior variance is very small, $s^2 \to 0$, then we have $\frac{\sigma^2}{ns^2+\sigma^2} \to 1$ while $\frac{ns^2}{ns^2+\sigma^2} \to 0$, yielding $\hat{\mu}_{MAP} \to m$. That is, if our prior is very concentrated, we essentially already know the answer, and can ignore the data.

4.1.15 Sequential (recursive) updating of $\hat{\Sigma}$

1. The covariance can be sequentially updated as follows

$$\begin{aligned}
\mathbf{C}_{n+1} &= \frac{1}{n}\sum_{k=1}^{n+1}(\mathbf{x}_k - \mathbf{m}_{n+1})(\mathbf{x}_k - \mathbf{m}_{n+1})^T \\
&= \frac{1}{n}\left[\sum_{k=1}^{n}(\mathbf{x}_k - \mathbf{m}_{n+1})(\mathbf{x}_k - \mathbf{m}_{n+1})^T + (\mathbf{x}_{n+1} - \mathbf{m}_{n+1})(\mathbf{x}_{n+1} - \mathbf{m}_{n+1})^T\right] \\
&= \frac{1}{n}\left[\sum_{k=1}^{n}(\mathbf{x}_k - \mathbf{m}_n)(\mathbf{x}_k - \mathbf{m}_n)^T - \frac{1}{(n+1)}(\mathbf{x}_{n+1} - \mathbf{m}_n)\sum_{k=1}^{n}(\mathbf{x}_k - \mathbf{m}_n)^T\right. \\
&- \frac{1}{(n+1)}\left(\sum_{k=1}^{n}(\mathbf{x}_k - \mathbf{m}_n)\right)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T + \frac{1}{(n+1)^2}\sum_{k=1}^{n}(\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n) \\
&+ \left.\frac{1}{n}\left((\mathbf{x}_{n+1} - \mathbf{m}_n) - \frac{1}{(n+1)}(\mathbf{x}_{n+1} - \mathbf{m}_n)\right)\left((\mathbf{x}_{n+1} - \mathbf{m}_n) - \frac{1}{(n+1)}(\mathbf{x}_{n+1} - \mathbf{m}_n)\right)\right] \\
&= \frac{1}{n}\left[(n-1)\mathbf{C}_n + \frac{n}{(n+1)^2}(\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T\right] \\
&+ \frac{1}{n}\left(\left(\frac{n}{n+1}\right)^2(\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T\right) \\
&= \frac{n-1}{n}\mathbf{C}_n + \left(\frac{1}{(n+1)^2} + \frac{n}{(n+1)^2}\right)(\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T \\
&= \frac{n-1}{n}\mathbf{C}_n + \frac{1}{(n+1)}(\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T
\end{aligned}$$

2. It takes $O(d^2)$ time per sequential update of $\mathbf{C}$ to compute the outer product and add it to the $d \times d$ matrix.

3. First we write the covariance update in a manner suitable for the rank-one update formula

$$\begin{aligned}
\mathbf{C}_{n+1} &= \frac{n-1}{n}\mathbf{C}_n + \frac{1}{(n+1)}(\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T \\
&= \frac{n-1}{n}\left[\mathbf{C}_n + \frac{n}{(n^2-1)}(\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T\right] \\
&= \frac{n-1}{n}(\mathbf{A} + \mathbf{x}\mathbf{x}^T)
\end{aligned}$$

where $\mathbf{A} = \mathbf{C}_n$ and $\mathbf{x} = \sqrt{\frac{n}{n^2-1}}(\mathbf{x}_{n+1} - \mathbf{m}_n)$. We then substitute into the matrix inversion lemma to get

$$\begin{aligned}
\mathbf{C}_{n+1}^{-1} &= \left[\frac{n-1}{n}(\mathbf{A} + \mathbf{x}\mathbf{x}^T)\right]^{-1} = \frac{n}{n-1}(\mathbf{A} + \mathbf{x}\mathbf{x}^T)^{-1} \\
&= \frac{1}{(n+1)}\left[\mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{x}\mathbf{x}^T\mathbf{A}^{-1}}{1 + \mathbf{x}^T\mathbf{A}^{-1}\mathbf{x}}\right] \\
&= \frac{n}{(n+1)}\left[\mathbf{C}_n^{-1} - \frac{\mathbf{C}_n^{-1}\sqrt{\frac{n}{n^2-1}}(\mathbf{x}_{n+1} - \mathbf{m}_n)\sqrt{\frac{n}{n^2-1}}(\mathbf{x}_{n+1} - \mathbf{m}_n)^T\mathbf{C}_n^{-1}}{1 + \sqrt{\frac{n}{n^2-1}} + (\mathbf{x}_{n+1} - \mathbf{m}_n)^T\mathbf{C}_n^{-1}\sqrt{\frac{n}{n^2-1}}(\mathbf{x}_{n+1} - \mathbf{m}_n)}\right] \\
&= \frac{n}{n-1}\left[\mathbf{C}_n^{-1} - \frac{\mathbf{C}_n^{-1}(\mathbf{x}_{n+1} - \mathbf{m}_n)(\mathbf{x}_{n+1} - \mathbf{m}_n)^T\mathbf{C}_n^{-1}}{\frac{n^2-1}{n} + (\mathbf{x}_{n+1} - \mathbf{m}_n)^T\mathbf{C}_n^{-1}(\mathbf{x}_{n+1} - \mathbf{m}_n)}\right]
\end{aligned}$$

4. It takes $O(d^2)$ to compute $\mathbf{u} = \mathbf{C}_n^{-1}(\mathbf{x}_{n+1} - \mathbf{m}_n)$, $O(d^2)$ to compute $\mathbf{C}_n^{-1} - \mathbf{u}\mathbf{u}^T$, and hence $O(d^2)$ time overall per update.

4.1.16 Likelihood ratio for Gaussians

For the general case, we have

$$\log\frac{p(\mathbf{x}|Y=1)}{p(\mathbf{x}|Y=0)} = \log\frac{|\Sigma_1|^{-\frac{1}{2}}\exp[-\frac{1}{2}(\mathbf{x}-\mu_1)^T\Sigma_1^{-1}(\mathbf{x}-\mu_1)]}{|\Sigma_0|^{-\frac{1}{2}}\exp[-\frac{1}{2}(\mathbf{x}-\mu_0)^T\Sigma_0^{-1}(\mathbf{x}-\mu_0)]} \tag{4.100}$$

If $\Sigma_1 = \Sigma_0 = \Sigma$, we have

$$\log \frac{p(\mathbf{x}|Y=1)}{p(\mathbf{x}|Y=0)} = \beta^T \mathbf{x} + \gamma \tag{4.101}$$

$$\beta \triangleq \Sigma^{-1}(\mu_1 - \mu_0) \tag{4.102}$$

$$\gamma \triangleq -\frac{1}{2}(\mu_1 - \mu_0)^T \Sigma^{-1}(\mu_1 + \mu_0) \tag{4.103}$$

If Σ is shared and diagonal, we have

$$\log \frac{p(\mathbf{x}|Y=1)}{p(\mathbf{x}|Y=0)} = \sum_{j=1}^{d} \frac{\mu_{1j} - \mu_{0j}}{\sigma_j^2} x_j - \frac{1}{2} \sum_{j=1}^{d} \frac{\mu_{1j}^2 + \mu_{0j}^2}{\sigma_j^2} \tag{4.104}$$

If Σ is shared and spherical we have

$$\log \frac{p(\mathbf{x}|Y=1)}{p(\mathbf{x}|Y=0)} = \frac{1}{\sigma^2} \sum_{j=1}^{d} (\mu_{1j} - \mu_{0j}) x_j - \frac{1}{2\sigma^2} \sum_{j=1}^{d} (\mu_{1j}^2 + \mu_{0j}^2) \tag{4.105}$$

4.1.17 LDA/QDA on height/weight data

I get the following error rates: LDA 0.1286, QDA 0.1190.

Here is my code.

Listing 4.2: discrimAnalysisHeightWeightDemo.m

```
%% Discriminative Analysis On the height weihgt data

rawdata = dlmread('heightWeightData.txt'); % comma delimited file
y = rawdata(:,1); % 1=male, 2=female
X = [rawdata(:,2) rawdata(:,3)]; % height, weight

model = discrimAnalysisFit(X, y, 'linear');
[yhat] = discrimAnalysisPredict(model, X);
errRateLDA = mean( (yhat ~= y))

model = discrimAnalysisFit(X, y, 'quadratic');
[yhat] = discrimAnalysisPredict(model, X);
errRateQDA = mean( (yhat ~= y))
```

4.1.18 Naive Bayes with mixed features

We have

$$p(y=c, x_1, x_2) = \pi(c) \mathrm{Ber}(x_1|\theta_c) \mathcal{N}(x_2|\mu_c, \sigma_c^2) \tag{4.106}$$

where

$$\mathrm{Ber}(x_1|\theta_c) = \theta_c^{I(x_1=1)} (1-\theta_c)^{I(x_1=1)} = 0.5^{I(x_1=1)} 0.5^{I(x_1=1)} = 0.5 \tag{4.107}$$

Thus feature 1 is irrelevant, so we have $p(y|x_1, x_2) = p(y|x_2)$ and $p(y|x_1) = p(y)$. We have

$$p(y=c, x_2=0) = \pi(c) \frac{1}{\sqrt{2\pi\sigma_c}} \exp(-\frac{1}{2\sigma_c^2}(x_2 - \mu_c)^2) \tag{4.108}$$

$$= \pi(c) \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}\mu_c^2) \tag{4.109}$$

$$\propto \pi(c) \exp(-\frac{1}{2}\mu_c^2) \tag{4.110}$$

We get the vector

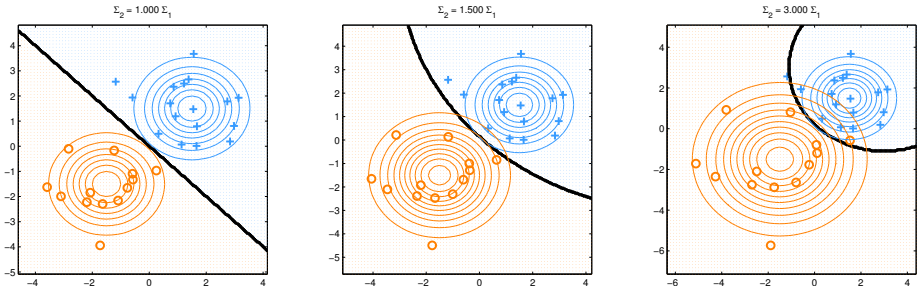
$$p(y, x_2=0) = [0.5, 0.25, 0.25] .* \exp([-0.5, 0, -0.5]) \tag{4.111}$$

$$= [0.5, 0.25, 0.25] .* \exp([-0.5, 0, -0.5]) \tag{4.112}$$

$$= [0.3033, 0.2500, 0.1516] \tag{4.113}$$

and

$$p(y|x_1, x_2) = [0.3033, 0.2500, 0.1516] ./ 0.7049 = [0.4302, 0.3547, 0.2151] \tag{4.114}$$



(a) (b) (c)

Figure 4.1: Discriminant analysis in which $\mathbf{\Sigma}_2 = k\mathbf{\Sigma}_1$. (a) $k = 1$ (b) $k = 1.5$ (c) $k = 3$ Produced by `discrimAnalysisSemiTiedDemo`.

We find

$$p(y|x_1, x_2) = p(y|x_2) \tag{4.115}$$

since x_1 is uninformative. We also find

$$p(y|x_1) = p(y) \tag{4.116}$$

for the same reason.

Below is some Matlab code to compute this, in case you are in any doubt. But as you can see from the above, you don't need Matlab to solve this simple problem!

Listing 4.3: nbMultiFeaturesSol.m

```
% nbMultiFeaturesSol

prior = [0.5 0.25 0.25];
theta = [0.5 0.5 0.5];
mu = [-1 0 1];
sigma = [1 1 1];
x1 = 0; x2 = 0;
for c=1:3
  if x1==1
    lik1(c) = theta(c);
  else
    lik1(c) = 1-theta(c);
  end
  lik2(c) = gausspdf(x2, mu(c), sigma(c)^2);
end
post12 = normalize(lik1 .* lik2 .* prior)
assert(approxeq(post12, normalize([0.5 0.25 0.25] .* [0.5 0.5 0.5] .* exp([-0.5 0 -0.5]))))
post2 = normalize(lik2  .* prior)
post1 = normalize(lik1  .* prior)
```

The output is as follows:

```
post12 =
    0.4302    0.3547    0.2151
post1 =
    0.5000    0.2500    0.2500
post2 =
    0.4302    0.3547    0.2151
```

4.1.19 Decision boundary for LDA with semi tied covariances

We have

$$p(y=0, \mathbf{x}|\boldsymbol{\theta}) = \pi_0(2\pi)^{-D/2}|\mathbf{\Sigma}_0|^{-\frac{1}{2}} \exp(-\frac{1}{2}\left[\mathbf{x}^T\mathbf{\Sigma}_0^{-1}\mathbf{x} - 2\boldsymbol{\mu}_0^T\mathbf{\Sigma}_0^{-1}\mathbf{x} + \boldsymbol{\mu}_0^T\mathbf{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right]) \tag{4.117}$$

$$= \underbrace{(2\pi)^{-D/2}|\mathbf{\Sigma}_0|^{-\frac{1}{2}}}_{A} \exp[\underbrace{\boldsymbol{\mu}_0^T\mathbf{\Sigma}_0^{-1}}_{\boldsymbol{\beta}_0^T}\mathbf{x} \underbrace{-\frac{1}{2}\boldsymbol{\mu}_0^T\mathbf{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + \log(\pi_0)}_{\gamma_0}] \exp[\underbrace{-\frac{1}{2}\mathbf{x}^T\mathbf{\Sigma}_0^{-1}\mathbf{x}}_{c(\mathbf{x})}] \tag{4.118}$$

and

$$p(y=1,\mathbf{x}|\boldsymbol{\theta}) = \pi_1|2\pi k\boldsymbol{\Sigma}_0|^{-\frac{1}{2}}\exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T k^{-1}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)) \quad (4.119)$$

$$= \pi_1(2\pi)^{-D/2}k^{-D/2}|\boldsymbol{\Sigma}_0|^{-\frac{1}{2}}\exp(-\frac{1}{2k}\left[\mathbf{x}^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} - 2\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_0^{-1}\mathbf{x} + \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_1\right]) \quad (4.120)$$

$$= A\exp[\underbrace{\frac{1}{k}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_0^{-1}}_{\boldsymbol{\beta}_1^T}\mathbf{x} \underbrace{-\frac{1}{2k}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_1 + \log(\pi_1) - \frac{D}{2}\log(k)}_{\gamma_1}]\exp[\frac{c(\mathbf{x})}{k}] \quad (4.121)$$

Hence

$$p(y=1|\mathbf{x},\boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1+c(\mathbf{x})/k)}{\exp(\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1+c(\mathbf{x})/k)+\exp(\boldsymbol{\beta}_0^T\mathbf{x}+\gamma_0+c(\mathbf{x}))} \quad (4.122)$$

$$= \frac{1}{1+\exp((\boldsymbol{\beta}_0-\boldsymbol{\beta}_1)^T\mathbf{x}+(\gamma_0-\gamma_1)+c(\mathbf{x})(1-1/k))} \quad (4.123)$$

Thus the decision boundaries are quadratic unless $k=1$. Furthermore, since $c(\mathbf{x})(1-1/k) < 0$ for $k > 1$, we see that increasing k is like increasing γ_1 which is like increasing π_1, so the decision region for class 1 grows larger, and the decision region for class 0 becomes a smaller and smaller ellipsoid. This is illustrated in Figure 4.1, which was produced by the code below.

Note: some people forgot that $c(\mathbf{x})$ is a function of $\mathbf{x}$. If you treat it as a constant, you would falsely conclude that the decision boundary is always linear.

Listing 4.4: discrimAnalysisSemiTiedDemo.m

```
ks = [1 1.5 3];
for trial=1:length(ks)
  k = ks(trial);
  model.mixweight  = [1 1]/2;
  model.classPrior = model.mixweight;
  model.mu         = [1.5 1.5 ; -1.5 -1.5]';
  model.Sigma(:,:,1) = eye(2);
  model.Sigma(:,:,2) = k*eye(2);
  model.type       = 'quadratic'; % not tied

  setSeed(3); nsamples = 30;
  colors = pmtkColors();
  xyRange = [-10 10 -10 10];

  [X, y] = mixGaussSample(model, nsamples);
  plotDecisionBoundary(X, y, @(Xtest)discrimAnalysisPredict(model, Xtest));
  for j = 1:2
    fn = @(x)gausspdf(x, model.mu(:, j), model.Sigma(:, :, j));
    plotContour(fn, xyRange, 'LineColor', colors{j});
  end
  title(sprintf('%s = %5.3f %s', '\Sigma_2', k, '\Sigma_1'))
  axis square
  fname = sprintf('discrimAnalysisSemiTied%d', trial)
  printPmtkFigure(fname)
end
```

4.1.20 Logistic regression vs LDA/QDA

1. GaussI $\leq$ LinLog. Both have logistic (sigmoid) posteriors $p(y|\mathbf{x},\mathbf{w}) = \sigma(y\mathbf{w}^T\mathbf{x})$, but LinLog is the logistic model which is trained to maximize $p(y|\mathbf{x},\mathbf{w})$. (GaussI may have high joint $p(y,\mathbf{x})$, but this does not necessarily mean $p(y|\mathbf{x})$ is high; LinLog can achieve the maximum of $p(y|\mathbf{x})$, so will necessarily do at least as well as GaussI.)

2. GaussX $\leq$ QuadLog. Both have logistic posteriors with quadratic features, but QuadLog is the model of this class maximizing the average log probabilities.

3. LinLog $\leq$ QuadLog. Logistic regression models with linear features are a subclass of logistic regression models with quadratic functions. The maximum from the superclass is at least as high as the maximum from the subclass.

4. GaussI $\leq$ QuadLog. Follows from above inequalities.

5. Although one might expect that higher log likelihood results in better classification performance, in general, having higher average log $p(y|\mathbf{x})$ does not necessarily translate to higher or lower classification error. For example, consider linearly separable data. We have L(linLog) > L(GaussI), since maximum likelihood logistic regression will set the weights to infinity, to maximize the probability of the correct labels (hence $p(y_i|\mathbf{x}_i,\hat{\mathbf{w}}) = 1$ for all i). However, we have R(linLog) =

R(gaussI), since the data is linearly separable. (The GaussI model may or may not set σ very small, resulting in possibly very large class conditional pdfs; however, the posterior over y is a discrete pmf, and can never exceed 1.)

As another example, suppose the true label is always 1 (as opposed to 0), but model M always predicts $p(y = 1|\mathbf{x}, M) = 0.49$. It will always misclassify, but it is at least close to the decision boundary. By contrast, there might be another model M' that predicts $p(y = 1|\mathbf{x}, M') = 1$ on even-numbered inputs, and $p(y = 1|\mathbf{x}, M') = 0$ on odd-numbered inputs. Clearly $R(M') = 0.5 < R(M) = 1$, but $L(M') = -\infty < L(M) = \log(0.49)$.

4.1.21 Gaussian decision boundaries

1. We solve

$$\frac{1}{\sqrt{(2\pi)}\sigma_1}\exp[-\frac{1}{2\sigma_1^2}(x-\mu_1)^2] = \frac{1}{\sqrt{(2\pi)}\sigma_2}\exp[-\frac{1}{2\sigma_2^2}(x-\mu_2)^2] \tag{4.124}$$

$$-\log\sigma_1 - \frac{1}{2\sigma_1^2}(x-\mu_1)^2 = -\log\sigma_2 - \frac{1}{2\sigma_2^2}(x-\mu_2)^2 \tag{4.125}$$

To simplify the algebra we substitute $\mu_1 = 0, \sigma_1^2 = 1, \mu_2 = 1$ and continue to grind away...

$$-\frac{1}{2}x^2 = -\log\sigma_2 - \frac{1}{2\sigma_2^2}(x-1)^2 \tag{4.126}$$

$$0 = -x^2 + \frac{1}{\sigma_2^2}x^2 - \frac{1}{\sigma_2^2} + 2\log\sigma_2 \tag{4.127}$$

Now let

$$0 = ax^2 + bx + c \tag{4.128}$$

$$a = \frac{1}{\sigma_2^2} - 1 \tag{4.129}$$

$$b = 0 \tag{4.130}$$

$$c = -\frac{1}{\sigma_2^2} + 2\log\sigma_2 \tag{4.131}$$

So

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \pm 3.3717 \tag{4.132}$$

The decision region is sketched in Figure 4.2. Note that R_2 is *discontinuous*.

Here is a way to solve this problem using Matlab's symbolic algebra toolbox (thanks to C J Hawkins).

```
syms x
mu1=0;
sigma1=1;
mu2=1;
sigma2=1000; % stnd deviation
double(solve(normpdf(x,mu1, sigma1)-normpdf(x,mu2, sigma2)))
ans =
   -3.7169
    3.7169
```

2. If $\sigma_2 = 1 = \sigma_1$, we know that the decision boundary is equi-distant between the means:

$$x^* = (\mu_1 + \mu_2)/2 = 0.5 \tag{4.133}$$

Hence $R_1 = \{x : x \leq x^*\}$, i.e., all points to the left of 0.5. Similarly $R_2 = \{x : x \geq x^*\}$, i.e., all points to the right of 0.5. In this case, R_2 is a single connected region.

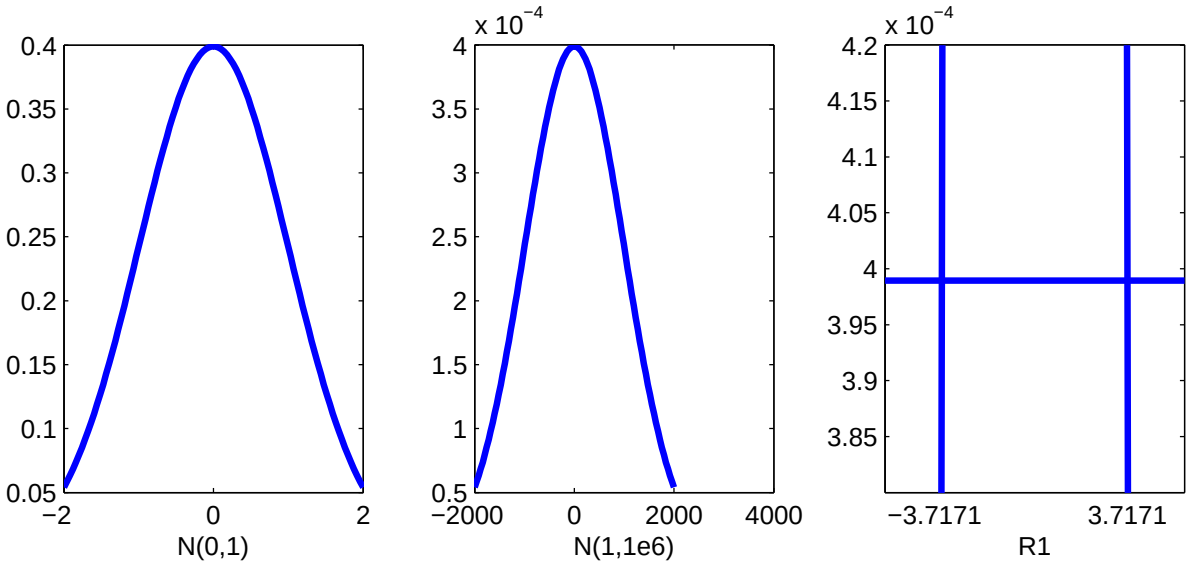


Figure 4.2: Class conditional densities $p(x|y=1) = \mathcal{N}(0,1)$ and $p(x|y=2) = \mathcal{N}(1, 10^6)$ and the corresponding decision region for class 1.

4.1.22 QDA with 3 classes

We just need to pick the most probable class. Here is some matlab code to compute the class posteriors.

Listing 4.5: :

```
mu1=[0;0];mu2=[1;1];mu3=[-1;1];
params.mu=[mu1 mu2 mu3];
params.Sigma(:,:,1)=[0.7 0;0 0.7];
params.Sigma(:,:,2)=[0.8 0.2;0.2 0.8];
params.Sigma(:,:,3)=[0.8 0.2;0.2 0.8];
params.Classprior=[1/3 1/3 1/3];
X1=[-0.5 0.5];
X2=[0.5 0.5];
post1=classify3gauss(X1,params)
post2=classify3gauss(X2,params)

function post=classify3gauss(Xtest,params)
Nclasses=length(params.Classprior)
for c=1:Nclasses
    lik(:,c) = mvnpdf(Xtest, params.mu(:, c)', params.Sigma(:,:, c));
end
classPrior = params.Classprior;
N = size(Xtest,1);
logjoint = log(lik) + repmat(log(classPrior(:)'), N, 1);
logpost = logjoint - repmat(logsumexp(logjoint,2), 1, Nclasses);
post = exp(logpost);
```

You get the following results:

$$P(Y=1|\vec{x_1}) = 0.46, P(Y=2|\vec{x_1}) = 0.145, P(Y=3|\vec{x_1}) = 0.39$$

$$R(\alpha_1|\vec{x_1}) = 0.53, R(\alpha_2|\vec{x_1}) = 0.85, R(\alpha_3|\vec{x_1}) = 0.60$$

(Here, $R(\alpha_1|\vec{x_1}) = 0.145 + 0.39 = 0.53$, etc.) So class 1 has minimal risk/ maximum posterior probability.

For the second vector:

$$P(Y=1|\vec{x_2}) = 0.45, P(Y=2|\vec{x_2}) = 0.46, P(Y=3|\vec{x_2}) = 0.09$$

$$R(\alpha_1|\vec{x_2}) = 0.55, R(\alpha_2|\vec{x_2}) = 0.54, R(\alpha_3|\vec{x_1}) = 0.91$$

Choose Class 2! (Although class 1 is very close : we might prefer the "reject" option if it was available, in this case.)

4.1.23 Scalar QDA

1. The MLE parameters are

$$\mu_m = \frac{67+79+71}{3} = 72.33 \tag{4.134}$$

$$\sigma_m^2 = \frac{(67-72.33)^2+(79-72.33)^2+(71-72.33)^2}{3} = 24.8889 \tag{4.135}$$

$$\sigma_m = 4.9889 \tag{4.136}$$

$$\mu_f = \frac{68+67+60}{3} = 65 \tag{4.137}$$

$$\sigma_f^2 = \frac{(68-65)^2+(67-65)^2+(60-65)^2}{3} = 12.667 \tag{4.138}$$

$$\sigma_f = 3.5590 \tag{4.139}$$

2. The predicted probability is given by

$$p_1 = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2\sigma_1^2}(x-\mu_1)^2} = 0.0798 \tag{4.140}$$

$$\tag{4.141}$$

$$p_2 = \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{1}{2\sigma_2^2}(x-\mu_2)^2} = 0.0162 \tag{4.142}$$

$$p = \frac{\pi_1 p_1}{\pi_1 p_1 + \pi_2 p_2} = 0.8312 \tag{4.143}$$

Here is my solution in Matlab

Listing 4.6: gaussClassifierSimpleScript.m

```
% gaussClassifierSimpleScript

X = [ 67  1;
      79   1;
      71   1;
      68    0 ;
      67   0;
      60    0];

male = find(X(:,2)==1);
female = find(X(:,2)==0);
mu(1) = mean(X(male,1));
mu(2) = mean(X(female,1));
sigma(1) = std(X(male,1),1);
sigma(2) = std(X(female,1),1);
N = size(X,1);
pc(1) = length(male) / N;
pc(2) = length(female) / N;

x=72;
p1 = normpdf(x,mu(1),sigma(1))
p1 = 1/sqrt(2*pi*sigma(1)^2)*exp(-0.5*(x-mu(1))^2/sigma(1)^2)
p2 = normpdf(x,mu(2),sigma(2))
p2 =  1/sqrt(2*pi*sigma(2)^2)*exp(-0.5*(x-mu(2))^2/sigma(2)^2)
prob = (pc(1)*p1)/(pc(1)*p1+pc(2)*p2)
```

Chapter 5

Bayesian statistics

5.1 Solutions

5.1.1 Proof that a mixture of conjugate priors is indeed conjugate

We now show that, if the prior is a mixture, so is the posterior:

$$
\begin{aligned}
p(\theta|\mathcal{D}) &= \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} && (5.1)\\
&= \frac{p(\mathcal{D}|\theta)\sum_k p(Z=k)p(\theta|Z=k)}{\int p(\mathcal{D}|\theta')\sum_{k'} p(Z=k')p(\theta'|Z=k')d\theta'} && (5.2)\\
&= \frac{\sum_k p(Z=k)p(\mathcal{D},\theta|Z=k)}{\sum_{k'} p(Z=k')\int p(\mathcal{D},\theta'|Z=k')d\theta'} && (5.3)\\
&= \frac{\sum_k p(Z=k)p(\theta|\mathcal{D},Z=k)p(\mathcal{D}|Z=k)}{\sum_{k'} p(Z=k')p(\mathcal{D}|Z=k')} && (5.4)\\
&= \sum_k \left[\frac{p(Z=k)p(\mathcal{D}|Z=k)}{\sum_{k'} p(Z=k')p(\mathcal{D}|Z=k')}\right] p(\theta|\mathcal{D},Z=k) && (5.5)\\
&= \sum_k p(Z=k|\mathcal{D})p(\theta|\mathcal{D},Z=k) && (5.6)
\end{aligned}
$$

where $p(Z=k)=\pi_k$ are the prior mixing weights, and $p(Z=k|\mathcal{D})$ are the posterior mixing weights given by

$$
p(Z=k|\mathcal{D}) = \frac{p(Z=k)p(\mathcal{D}|Z=k)}{\sum_{k'} p(Z=k')p(\mathcal{D}|Z=k')} \quad (5.7)
$$

5.1.2 Optimal threshold on classification probability

1. Minimizing the expected is equivalent to minimizing the risk (posterior expected loss) which is defined by:

$$
R(\alpha_i|\vec{x}) = \lambda_{ii}P(Y=i|\vec{x}) + \sum_{i\neq j}\lambda_{ij}P(Y=j|\vec{x}) \quad (5.8)
$$

In addition we know that $P(Y=0|\vec{x})$ and $P(Y=1|\vec{x})$ are complementary events therefore $p_0 = 1-p_1$ Let's consider the general case where the loss matrix is given by:

$$
\lambda = \begin{pmatrix} \ell_{00} & \ell_{01} \\ \ell_{10} & \ell_{11} \end{pmatrix} \quad (5.9)
$$

The risk functions are given by:

$$
R(\alpha_0|\vec{x}) = \ell_{00}(1-p_1) + \ell_{01}p_1 \quad (5.10)
$$

$$
R(\alpha_1|\vec{x}) = \ell_{10}(1-p_1) + \ell_{11}p_1 \quad (5.11)
$$

Using the given loss matrix, we will choose $Y=0$ if

$$
\begin{aligned}
R(\alpha_0|\vec{x}) &< R(\alpha_1|\vec{x}) && (5.12)\\
\ell_{00}(1-p_1) + \ell_{01}p_1 &< \ell_{10}(1-p_1) + \ell_{11}p_1 && (5.13)
\end{aligned}
$$

If $\ell_{00} = \ell_{11} = 0$, then

$$\begin{aligned} \ell_{01} p_1 &< \ell_{10}(1 - p_1) && (5.14) \\ p_1(\ell_{01} + \ell_{10}) &< \ell_{10} && (5.15) \\ p_1 &< \frac{\ell_{10}}{\ell_{10} + \ell_{01}} = \theta && (5.16) \end{aligned}$$

2. From the above example, we see that if $\ell_{10} = 1$ and $\ell_{01} = 9$, we get $\theta = 0.1$, i.e., the following loss matrix will give a threshold of 0.1

predicted label $\hat{y}$	true label y	
	0	1
0	0	9
1	1	0

We can verify this as follows

$$R(\alpha_0|\vec{x}) < R(\alpha_1|\vec{x}) \Rightarrow 9p_1 < (1 - p_1) \Rightarrow 10p_1 < 1 \Rightarrow p_1 < 1/10 = \theta$$

5.1.3 Reject option in classifiers

1. () We have to choose between rejecting, with risk λ_r, and choosing the most probable class, $j_{max} = \arg\max_j p(Y = j|\mathbf{x})$, which has risk

$$\lambda_s \sum_{j \neq j_{max}} p(Y = j|\mathbf{x}) = \lambda_s(1 - p(Y = j_{max}|\mathbf{x})) \quad (5.17)$$

Hence we should pick j_{max} if

$$\begin{aligned} \lambda_r &\geq \lambda_s(1 - p(Y = j_{max}|\mathbf{x})) && (5.18) \\ \frac{\lambda_r}{\lambda_s} &\geq (1 - p(Y = j_{max}|\mathbf{x})) && (5.19) \\ p(Y = j_{max}|\mathbf{x}) &\geq 1 - \frac{\lambda_r}{\lambda_s} && (5.20) \end{aligned}$$

otherwise we should reject.

For completeness, we should prove that when we decide to choose a class (and not reject), we always pick the most probable one. If we choose a non-maximal category $k \neq j_{max}$, the risk is

$$\lambda_s \sum_{j \neq k} p(Y = j|\mathbf{x}) = \lambda_s(1 - p(Y = k|\mathbf{x})) \geq \lambda_s(1 - p(Y = j_{max}|\mathbf{x})) \quad (5.21)$$

which is always bigger than picking j_{max}.

2. () If $\lambda_r/\lambda_s = 0$, there is no cost to rejecting, so we always reject. As $\lambda_r/\lambda_s \to 1$, the cost of rejecting increases. We find $p(Y = j_{max}|\mathbf{x}) \geq 1 - \frac{\lambda_r}{\lambda_s}$ is always satisfied, so we always accept the most probable class.

5.1.4 More reject options

1. For the first posterior, it is straight forward to calculate the risk for each function:

$$P(Y = 1|\vec{x}) = 0.2 \Rightarrow P(Y = 0|\vec{x}) = 0.8$$

$$R(\alpha_0|\vec{x}) = 10(0.2) = 2, R(\alpha_1|\vec{x}) = 10(0.8) = 8, R(\alpha_r|\vec{x}) = 3(0.2) + 3(0.8) = 3$$

We choose $Y = 0$, since it has the smallest risk.

2. For the second belief state

$$P(Y = 1|\vec{x}) = 0.4 \Rightarrow P(Y = 0|\vec{x}) = 0.6$$

$$R(\alpha_0|\vec{x}) = 10(0.4) = 4, R(\alpha_1|\vec{x}) = 10(0.6) = 6, R(\alpha_r|\vec{x}) = 3(0.4) + 3(0.6) = 3$$

In this case we choose the reject option.

3. In general, we will choose $Y = 0$ if

$$R(\alpha_0|\vec{x}) < R(\alpha_1|\vec{x})$$

$$10p_1 < 10(1 - p_1)$$

$$10p_1 + 10p_1 < 10$$

$$p_1 < 0.5$$

and if

$$R(\alpha_0|\vec{x}) < R(\alpha_r|\vec{x})$$

$$10p_1 < 3p_1 + 3(1 - p_1)$$

$$7p_1 + 3p_1 < 3$$

$$p_1 < 0.3$$

we take

$$\theta_0 = \min\Big(0.5, 0.3\Big) = 0.3$$

Similarly, we choose $Y = 1$ if

$$R(\alpha_0|\vec{x}) \geq R(\alpha_1|\vec{x})$$

$$10p_1 \geq 10 - 10p_1$$

$$p_1 \geq .5$$

and if

$$R(\alpha_1|\vec{x}) < R(\alpha_r|\vec{x})$$

$$10 - 10p_1 < 3$$

$$p_1 > 0.7$$

we take

$$\theta_1 = \max\Big(0.5, 0.7\Big) = 0.7$$

Similarly, we choose Reject option if

$$R(\alpha_r|\vec{x}) < R(\alpha_0|\vec{x})$$

$$3 < 10p_1$$

$$p_1 > 0.3$$

or if

$$R(\alpha_r|\vec{x}) < R(\alpha_1|\vec{x})$$

$$3 < 10 - 10p_1$$

$$p_1 < 0.7$$

i.e if $\theta_0 = 0.3 < p_1 < 0.7 = \theta_1$

5.1.5 Newsvendor problem

$$
\begin{aligned}
E\pi(Q) &= \int_Q^\infty (P-C)Qf(D)dD + \int_0^Q (P-C)Df(D) - \int_0^Q C(Q-D)f(D)dD && (5.22)\\
&= (P-C)Q(1-F(Q)) + P\int_0^Q Df(D) - CQ\int_0^Q f(D)dD && (5.23)\\
&= (P-C)Q(1-F(Q)) + P\int_0^Q Df(D) - CQF(Q) && (5.24)
\end{aligned}
$$

Hence

$$
\begin{aligned}
\frac{d}{dQ}E\pi(Q) &= (P-C)(1-F(Q)) - (P-C)Qf(Q) + PQf(Q) - CF(Q) - CQf(Q) && (5.25)\\
&= (P-C) - PF(Q) && (5.26)\\
&= 0 && (5.27)
\end{aligned}
$$

So

$$
F(Q) = \frac{P-C}{P} \tag{5.28}
$$

5.1.6 Bayes factors and ROC curves

$p(H_1|D)$ is a monotonically increasing function of B, and therefore cannot affect the shape of the ROC curve. This makes sense intuitively, since, in the two class case, it should not matter whether we threshold the ratio $p(H_1|D)/p(H_0|D)$, or the posterior, $p(H_1|D)$, since they contain the same information, just measured on different scales.

5.1.7 Bayes model averaging helps predictive accuracy

See (Madigan and Raftery 1994).

5.1.8 MLE and model selection for a 2d discrete distribution

1. The joint distribution is $p(x, y|\boldsymbol{\theta}) = p(x|\theta_1)p(y|x, \theta_2)$:

	$y=0$	$y=1$
$x=0$	$(1-\theta_1)\theta_2$	$(1-\theta_1)(1-\theta_2)$
$x=1$	$\theta_1(1-\theta_2)$	$\theta_1\theta_2$

2. The log likelihood is

$$
\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_i \log p(x_i|\theta_1) + \sum_i \log p(y_i|x_i, \theta_2) \tag{5.29}
$$

Hence we can optimize each term separately. For θ_1, we have

$$
\hat{\theta}_1 = \frac{\sum_i I(x_i = 1)}{n} = \frac{N(x=1)}{N} = \frac{4}{7} = 0.5714 \tag{5.30}
$$

For θ_2, we have

$$
\hat{\theta}_2 = \frac{\sum_i I(x_i = y_i)}{n} = \frac{N(x=y)}{N} = \frac{4}{7} \tag{5.31}
$$

The likelihood is

$$
\begin{aligned}
p(\mathcal{D}|\hat{\boldsymbol{\theta}}, M_2) &= (\frac{4}{7})^{N(x=1)}(\frac{3}{7})^{N(x=0)}(\frac{4}{7})^{N(x=y)}(\frac{3}{7})^{N(x\neq y)} && (5.32)\\
&= (\frac{4}{7})^4(\frac{3}{7})^4(\frac{4}{7})^4(\frac{3}{7})^4 && (5.33)\\
&= (\frac{4}{7})^8(\frac{3}{7})^6 \approx 7.04 \times 10^{-5} && (5.34)
\end{aligned}
$$

3. The table of joint counts is

	$y=0$	$y=1$
$x=0$	2	1
$x=1$	2	2

We can think of this as a multinomial distribution with 4 states. Normalizing the counts gives the MLE:

	$y=0$	$y=1$
$x=0$	2/7	1/7
$x=1$	2/7	2/7

The likelihood is

$$p(\mathcal{D}|\hat{\boldsymbol{\theta}}, M_4) = \theta_{00}^{N(x=0,y=0)}\theta_{01}^{N(x=0,y=1)}\theta_{10}^{N(x=1,y=0)}\theta_{11}^{N(x=1,y=1)} = (\frac{2}{7})^2(\frac{1}{7})^1(\frac{2}{7})^2(\frac{2}{7})^2 \tag{5.35}$$

$$= (\frac{2}{7})^6(\frac{1}{7})^1 \approx 7.77 \times 10^{-5} \tag{5.36}$$

Thus is higher than the previous likelihood, because the model has more parameters.

4. For M_4, when we omit case 7, we will have $\hat{\theta}_{01} = 0$, so $p(x_7, y_7|m_4, \hat{\theta}) = 0$, so $L(m_4) = -\infty$. However, $L(m_2)$ will be finite, since all counts remain non zero when we leave out a single case. Hence CV will prefer M_2, since M_4 is overfitting.

5. The BIC score is

$$BIC(m) = \log p(\mathcal{D}|\hat{\boldsymbol{\theta}}, m) - \frac{\text{dof}(m)}{2}\log n \tag{5.37}$$

where $n = 7$. For M_2, we have $\text{dof} = 2$, so

$$BIC(m_2) = 8\log(\frac{4}{7}) + 6\log(\frac{3}{7}) - \frac{2}{2}\log 7 = -11.5066 \tag{5.38}$$

For M_4, we have $\text{dof} = 3$ because of the sum-to-one constraint, so

$$BIC(m_4) = 6\log(\frac{2}{7}) + 1\log(\frac{1}{7}) - \frac{3}{2}\log 7 = -12.3814 \tag{5.39}$$

So BIC also prefers m_2.

5.1.9 Posterior median is optimal estimate under L1 loss

To prove this, we expand the posterior expected loss as follows:

$$\rho(a|\mathbf{x}) = E_{\theta|x}|\theta - a| = \int_{\theta \geq a}(\theta - a)p(\theta|x)d\theta + \int_{\theta \leq a}(a - \theta)p(\theta|x)d\theta \tag{5.40}$$

$$= \int_a^\infty (\theta - a)p(\theta|x)d\theta + \int_{-\infty}^a (a - \theta)p(\theta|x)d\theta \tag{5.41}$$

Now recall the rule to differentiate under the integral sign (Equation **??**), repeated below for conv

$$\frac{d}{da}\int_{A(a)}^{B(a)}\phi(a,\theta)d\theta = \int_{A(a)}^{B(a)}\phi'(a,\theta)d\theta + \phi(a,B(a))B'(a) + \phi(a,A(a))A'(a) \tag{5.42}$$

where $\phi'(a,\theta) = \frac{d}{da}\phi(a,\theta)$. Applying this to the first integral in Equation 5.41, with $A(a) = a$, $B(a) = \infty$, $\phi(a,\theta) = (\theta - a)p(\theta|x)dy$, we have

$$\int_a^\infty (\theta - a)p(\theta|x)d\theta = \int_a^\infty -p(\theta|x)d\theta + 0 + 0 \tag{5.43}$$

Analogously, one can show

$$\int_{-\infty}^a (a - \theta)p(\theta|x)d\theta = \int_{-\infty}^a p(\theta|x)d\theta \tag{5.44}$$

Hence

$$\rho'(a|\mathbf{x}) = -\int_a^\infty p(\theta|x)d\theta + \int_{-\infty}^a p(\theta|x)d\theta \tag{5.45}$$

$$= -P(\theta \geq a|x) + P(\theta \leq a|x) = 0 \tag{5.46}$$

So the value of a that makes $\rho'(a|\mathbf{x}) = 0$ satisfies

$$P(\theta \geq a|\mathbf{x}) = P(\theta \leq a|\mathbf{x}) \tag{5.47}$$

Hence the optimal a is the posterior median.

5.1.10 Decision rule for trading off FPs and FNs

Let $y_1 = p(y = 1|\mathbf{x})$. We have

$$\frac{y_1}{1 - y_1} \geq \frac{L_{FP}}{L_{FN}} = c \tag{5.48}$$

$$y_1 \geq c - y_1 c \tag{5.49}$$

$$y_1 \geq \frac{c}{1 + c} \tag{5.50}$$

Chapter 6

Frequentist statistics

6.1 Solutions

6.1.1 Pessimism of LOOCV

The best misclassification rate is clearly 50%. Now consider using LOOCCV. Suppose $N_1 = N_2 = 2$, so the data is (1,1,2,2). Consider a training fold containing (1,1,2) and a testing fold containing (2). The way to minimize the training error on the training fold is to predict $\hat{y} = 1$, but this encurs 100% error rate on the test fold. This will be true no matter which train/test fold we consider, since every training fold will always have one class being the majority label, and the testing fold having the opposite label. A classifier that minimizes its error rate on the training folds will always pick the most probable class (since the features are irrelevant). Hence the error rate as estimated by CV is estimated as 100%.

6.1.2 James Stein estimator for Gaussian means

1. We have

$$m_0 = \overline{y} = 1527.5, \;\; s^2 = 1878.6, \;\; \tau_0^2 = 1378.6 \tag{6.1}$$

2. The posterior mean and variance is

$$E[\theta_i] = m_0 + (1 - \lambda_0)(y_i - m_0), \;\; \text{var}\,[\theta_i] = (1 - \lambda_0)\sigma^2 \tag{6.2}$$

where

$$\lambda_0 = \frac{\sigma^2}{\sigma^2 + \tau_0^2} = 0.266 \tag{6.3}$$

Hence

Listing 6.1: :

```
postMean = 1510.99       1527.87       1554.29       1505.85       1580.70       1485.30
postVar =  366.92
```

3. The 95% credible interval is

$$E[\theta_i] \pm 1.96\sqrt{\text{Var}[\theta_i]} \tag{6.4}$$

Hence

Listing 6.2: :

```
1473.44       1490.32       1516.74       1468.31       1543.16       1447.76
1548.53       1565.41       1591.83       1543.40       1618.25       1522.85
```

The posterior variance is an underestimate of the uncertainty, since we are using a plug-in estimate for m and τ, so the interval is too small. (Also, we are implicitly conditioning on the fact that the Gaussian model assumption is correct.)

4. If we reduce σ^2, we trust the data more, and hence shrink less towards m_0. So the posterior means will be closer to the raw data. We can easily rerun the code with $\sigma^2 = 1$ to check this:

Listing 6.3: jamesSteinDemo.m

```
y=[1505, 1528, 1564, 1498, 1600, 1470];
sigma2 = 500;
%sigma2=1;
ybar=mean(y);
s2 = var(y,1);
t2 = s2-sigma2;

lambda = sigma2/(sigma2+t2);
postMean = ybar + (1-lambda).*(y-ybar)
postVar = (1-lambda)* sigma2

s = sqrt(postVar);
credint = [postMean - 1.96*s; postMean + 1.96*s]
```

6.1.3 $\hat{\sigma}^2_{MLE}$ is biased

Because the variance of any random variable R is given by $var(R) = E[R^2] - (E[R])^2$, the expected value of the square of a Gaussian random variable X_i with mean μ and variance σ^2 is $E[X_i^2] = var(X_i) + (E[X_i])^2 = \sigma^2 + \mu^2$.

$$\begin{aligned}
E_{X_1,\ldots,X_n\sim\mathcal{N}(\mu,\sigma)}[\sigma^2(X_1,\ldots,X_n)] &= E[\frac{1}{n}\sum_{i=1}^{n}(X_i - \frac{\sum_{j=1}^{n} X_j}{n})^2] \\
&= \frac{1}{n}\sum_{i=1} nE[(X_i - \frac{\sum_{j=1}^{n} X_j}{n})^2] \\
&= \frac{1}{n}\sum_{i=1} nE[(X_i - \frac{\sum_{j=1}^{n} X_j}{n})(X_i - \frac{\sum_{j=1}^{n} X_j}{n})] \\
&= \frac{1}{n}\sum_{i=1} nE[X_i^2 - \frac{2}{n}X_i\sum_{j=1}^{n} X_j + \frac{1}{n^2}\sum_{j=1}^{n}\sum_{k=1}^{n} X_jX_k] \\
&= \frac{1}{n}\sum_{i=1} nE[X_i^2] - \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} E[X_iX_j] + \frac{1}{n^3}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}[X_jX_k]
\end{aligned}$$

Consider the two summations $\sum_{i=1}^{n}\sum_{j=1}^{n} E[X_iX_j]$ and $sum_{j=1}^{n}\sum_{k=1}^{n} X_jX_k]$. Of the n^2 terms in each of these summations, n of them satisfy $i = j$ or $j = k$, so these terms are of the form $E[X_i^2]$. By linearity of expectation, these terms contribute $nE[X_i^2]$ to the sum. The remaining $n^2 - n$ terms are of the form $E[X_iX_j]$ or $E[X_jX_k]$ for $i \neq j$ or $j \neq k$. Because the X_i are independent samples, it follows from linearity of expectation that these terms contribute $(n^2 - n)E[X_i]E[X_j]$ to the summation.

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} E[X_iX_j] &= \sum_{j=1}^{n}\sum_{k=1}^{n} E[X_jX_k] \\
&= nE[X_i^2] + (n^2 - n)E[X_i][X_j] \\
&= n(\sigma^2 + \mu^2) + (n^2 - n)\mu\mu = n\sigma^2 + n\mu^2 + n^2\mu^2 - n\mu^2 \\
&= n\sigma^2 + n^2\mu^2
\end{aligned}$$

$$\begin{aligned}
&E_{X_1,\ldots,X_n\sim\mathcal{N}(\mu,\sigma)}[\sigma^2(X_1,\ldots,X_n)] = \\
&\quad \frac{1}{n}\sum_{i} = 1^n(\sigma^2 + \mu^2) - \frac{2}{n^2}(n\sigma^2 + n^2\mu^2) + \frac{1}{n^3}\sum_{i=1}^{n}(n\sigma^2 + n^2\mu^2) \\
&= \frac{1}{n}(n\sigma^2 + n\mu^2) - 2\frac{\sigma^2}{n} - 2\mu^2 + \frac{1}{n^3}(n^2\sigma^2 + N^3\mu^2) \\
&= \sigma^2 + \mu^2 - 2\frac{\sigma^2}{n} - 2\mu^2 + \frac{\sigma^2}{n} + \mu^2 \\
&= \sigma^2 - \frac{\sigma^2}{n} = \frac{n-1}{n}\sigma^2
\end{aligned}$$

Since the expected value of $\hat{\sigma^2}(X_1,\ldots,X_n)$ is not equal to the actual variance σ^2, σ^2 is not an unbiased estimator. In fact, the maximum likelihood estimator tends to underestimate the variance. This is not surprising: consider the case of only a single

sample: we will never detect any variance. If there are multiple samples, we will detect variance, but since our estimate for the mean will tend to be shifted from the true mean in the direction of our samples, we will tend to underestimate the variance.

6.1.4 Estimation of σ^2 when μ is known

We can re-do the derivation of $\hat{\sigma}^2_{MLE}$, but instead of taking the maximum over μ , we just use the known μ. We get:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

In calculating the expected value of the estimator, note that we get exactly the definition of a variance of a random variable:

$$\begin{aligned} E_{X_1,\ldots,X_n\sim\mathcal{N}(\mu,\sigma)}[\hat{\sigma}^2(X_1, X_2, \ldots, X_n) &= \frac{1}{n}\sum_{i=1}^{n} E_{X_i\sim\mathcal{N}(\mu,\sigma)}[(X_i - \mu)^2] \\ &= \frac{1}{n}\sum_{i=1}^{n} Var[\mathcal{N}(\mu,\sigma)] \\ &= \frac{1}{n}\sum_{i=1}^{n} \sigma^2 \\ &= \sigma^2 \end{aligned}$$

So the estimator is unbiased.

Chapter 7

Linear regression

7.1 Solutions

7.1.1 Behavior of training set error with increasing sample size

For simple models, the training error will decrease as we increase N, as expected. But for complex models, the training error will *increase* to some plateau as N increases. The reason is this: initially the model is sufficiently powerful to simply memorize the training data, but as we are given more examples, it becomes harder to fit them all perfectly. In either case, eventually the error on the training set will match the error on the test set.

7.1.2 Multi-output linear regression

We have

$$\hat{\mathbf{W}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{Y} \tag{7.1}$$

$$\mathbf{\Phi}^T = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{7.2}$$

$$\mathbf{\Phi}^T\mathbf{\Phi} = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} \tag{7.3}$$

$$\mathbf{Y} = \begin{pmatrix} -1 & -1 \\ -1 & -2 \\ -2 & -1 \\ 1 & 1 \\ 1 & 2 \\ 2 & 1 \end{pmatrix} \tag{7.4}$$

$$\mathbf{\Phi}^T\mathbf{Y} = \begin{pmatrix} -4 & -4 \\ 4 & 4 \end{pmatrix} \tag{7.5}$$

$$(\mathbf{\Phi}^T\mathbf{\Phi})^{-1} = (1/3)\mathbf{I}_2 \tag{7.6}$$

$$\hat{\mathbf{W}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{Y} = \begin{pmatrix} -4/3 & -4/3 \\ 4/3 & 4/3 \end{pmatrix} \tag{7.7}$$

7.1.3 Centering and ridge regression

Suppose $\mathbf{X}$ is centered, so $\overline{\mathbf{x}} = 0$. Then

$$J(\mathbf{w}, w_0) = (\mathbf{y} - \mathbf{X}\mathbf{w} - w_0\mathbf{1})^T(\mathbf{y} - \mathbf{X}\mathbf{w} - w_0\mathbf{1}) + \lambda\mathbf{w}^T\mathbf{w} \tag{7.8}$$

$$= \mathbf{y}^T\mathbf{y} + \mathbf{w}^T(\mathbf{X}^T\mathbf{X})\mathbf{w} - 2\mathbf{y}^T(\mathbf{X}\mathbf{w}) + \lambda\mathbf{w}^T\mathbf{w} + \left(-2w_0\mathbf{1}^T\mathbf{y} + 2w_0\mathbf{1}^T\mathbf{X}\mathbf{w} + w_0\mathbf{1}^T\mathbf{1}w_0\right) \tag{7.9}$$

Consider the terms in brackets:

$$w_0\mathbf{1}^T\mathbf{y} = w_0 n\overline{y} \tag{7.10}$$

$$w_0\mathbf{1}^T\mathbf{X}\mathbf{w} = w_0\sum_i \mathbf{x}_i^T\mathbf{w} = n\overline{\mathbf{x}}^T\mathbf{w} = 0 \tag{7.11}$$

$$w_0\mathbf{1}^T\mathbf{1}w_0 = nw_0^2 \tag{7.12}$$

Optimizing wrt w_0 we find

$$\frac{\partial}{\partial w_0} J(\mathbf{w}, w_0) = -2n\overline{y} + 2nw_0 = 0 \tag{7.13}$$

$$\hat{w}_0 = \overline{y} \tag{7.14}$$

Optimizing wrt $\mathbf{w}$ we find

$$\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}, \hat{w}_0) = [2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y}] + 2\lambda\mathbf{w} = 0 \tag{7.15}$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{7.16}$$

7.1.4 MLE for σ^2 for linear regression

Just solve $\frac{\partial}{\partial \sigma^2} NLL(\hat{\mathbf{w}}, \sigma^2) = 0$ and do some algebra.

7.1.5 MLE for the offset term in linear regression

First solve $\frac{\partial}{\partial w_0} RSS(\mathbf{w}, w_0) = 0$. Substituting in we get

$$RSS(\mathbf{w}, \hat{w}_0) = \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\mathbf{w} - \overline{y} + \overline{\mathbf{x}}^T\mathbf{w})^2 = \sum_{i=1}^{N}((y_i - \overline{y}) - (\mathbf{x}_i - \overline{\mathbf{x}})^T\mathbf{w})^2 \tag{7.17}$$

$$= \sum_{i=1}^{N}(y_i^c - (\mathbf{x}_i^c)^T\mathbf{w})^2 = (\mathbf{y}_c - \mathbf{X}_c\mathbf{w})^T(\mathbf{y}_c - \mathbf{X}_c\mathbf{w}) \tag{7.18}$$

We can solve this by OLS.

7.1.6 MLE for simple linear regression

Simple calculus.

7.1.7 Sufficient statistics for online linear regression

1. From Equation **??** we have $w_1 = C_{xy}/C_{xx}$.
2. From Equation **??** we have $w_0 = \overline{y} - w_1\overline{x}$, where w_1 needs C_{xy} and C_{xx}.
3. We have

$$\overline{x}^{(n+1)} = \frac{1}{n+1}\left(n\overline{x}^{(n)} + x_{n+1}\right) \tag{7.19}$$

$$= \frac{(n+1)\overline{x}^{(n)} - \overline{x}^{(n)}}{n+1} + \frac{1}{n+1}x_{n+1} \tag{7.20}$$

$$= \overline{x}^{(n)} - \frac{1}{n+1}\overline{x}^{(n)} + \frac{1}{n+1}x_{n+1} \tag{7.21}$$

$$= \overline{x}^{(n)} + \frac{1}{n+1}(x_{n+1} - \overline{x}^{(n)}) \tag{7.22}$$

4. We have

$$C_{xy}^{(n)} = \frac{1}{n}\left[(\sum_{i=1}^{n} x_i y_i) + (\sum_{i=1}^{n} \overline{x}^{(n)}\overline{y}^{(n)}) - \overline{x}^{(n)}(\sum_{i=1}^{n} y_i) - \overline{y}^{(n)}(\sum_{i=1}^{n} x_i)\right] \tag{7.23}$$

$$= \frac{1}{n}\left[(\sum_{i=1}^{n} x_i y_i) + n\overline{x}^{(n)}\overline{y}^{(n)} - \overline{x}^{(n)}n\overline{y}^{(n)} - \overline{y}^{(n)}n\overline{x}^{(n)}\right] \tag{7.24}$$

$$= \frac{1}{n}\left[(\sum_{i=1}^{n} x_i y_i) - n\overline{x}^{(n)}\overline{y}^{(n)}\right] \tag{7.25}$$

Hence

$$\sum_{i=1}^{n} x_i y_i = nC_{xy}^{(n)} + n\overline{x}^{(n)}\overline{y}^{(n)} \tag{7.26}$$

and

$$C_{xy}^{(n+1)} = \frac{1}{n+1}\left[x_{n+1}y_{n+1} + nC_{xy}^{(n)} + n\overline{x}^{(n)}\overline{y}^{(n)} - (n+1)\overline{x}^{(n+1)}\overline{y}^{(n+1)}\right] \tag{7.27}$$

7.1.8 Bayesian linear regression in 1d with known σ^2

1. The prior has the form

$$p(\mathbf{w}) = \mathcal{N}\left(\mathbf{w}|\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}\infty & 0\\0 & 1\end{pmatrix}\right) \tag{7.28}$$

2. The posterior has the form

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \sigma^2) = \mathcal{N}(\mathbf{w}|\mathbf{w}_N, \mathbf{V}_N) \tag{7.29}$$

$$\mathbf{w}_N = \mathbf{V}_N\mathbf{V}_0^{-1}\mathbf{w}_0 + \frac{1}{\sigma^2}\mathbf{V}_N\mathbf{X}^T\mathbf{y} \tag{7.30}$$

$$\mathbf{V}_N^{-1} = \mathbf{V}_0^{-1} + \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X} \tag{7.31}$$

When computing this, we must use

$$\mathbf{V}_0^{-1} = \begin{pmatrix}0 & 0\\0 & 1\end{pmatrix} \tag{7.32}$$

The marginal has the form

$$p(w_1|\mathcal{D}, \sigma^2) = \mathcal{N}(w_1|\mathbf{w}_N(2), \mathbf{V}_N(2,2)) \tag{7.33}$$

$$= \mathcal{N}(w_1|0.0426514132, 0.0000114230) \tag{7.34}$$

3. The 95% CI is $(0.0360, 0.0493)$.

The code to compute this is shown below.

Listing 7.1: Listing of `bayesLinregBolstad`

```
x = [94,96,94,95,104,106,108,113,115,121,131];
y = [0.47, 0.75, 0.83, 0.98, 1.18, 1.29, 1.40, 1.60, 1.75, 1.90, 2.23];
y = y(:);
N=length(y)
X=[ones(N,1) x(:)];
w = X\y
yhat = X*w;
SSR = sum((yhat - y).^2)
sigma2 = SSR/(N-2)

w0 = zeros(2,1);
V0 = diag(inf, 1);
V0inv = diag(0, 1);
VN = inv(V0inv + (1/sigma2)*X'*X);
wvar = VN(2,2);
fprintf('post var(w1) = %10.10f\n', wvar) % 0.0000114230
wN = VN*V0inv*w0 + (1/sigma2)*VN*X'*y;
wmean = wN(2);
fprintf('post mean(w1) = %10.10f\n', wmean) % 0.0426514132

alpha = 0.05;
CI = [norminv(alpha/2, wmean, sqrt(wvar)), norminv(1-alpha/2, wmean, sqrt(wvar))]
```

7.1.9 Generative model for linear regression

1. We have

$$\hat{\mathbf{\Sigma}}_{XX} = \frac{1}{n}\sum_i (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T \tag{7.35}$$

$$= \frac{1}{n}(\mathbf{X} - \bar{\mathbf{X}})^T(\mathbf{X} - \bar{\mathbf{X}}) \tag{7.36}$$

Similarly

$$\hat{\mathbf{\Sigma}}_{XY} = \frac{1}{n}\sum_i (\mathbf{x}_i - \overline{\mathbf{x}})(y_i - \overline{y})^T \tag{7.37}$$

$$= \frac{1}{n}(\mathbf{X} - \bar{\mathbf{X}})^T(\mathbf{y} - \bar{\mathbf{y}}\mathbf{1}) \tag{7.38}$$

Hence

$$E[Y|\mathbf{x}] = \boldsymbol{\mu}_Y + \mathbf{\Sigma}_{YX}\mathbf{\Sigma}_{XX}^{-1}(y - \mu_Y) \tag{7.39}$$

$$= w_0 + \mathbf{w}^T\mathbf{y} \tag{7.40}$$

$$\mathbf{w} = \mathbf{\Sigma}_{XX}^{-1}\mathbf{\Sigma}_{YX}^T = \left((\mathbf{X} - \bar{\mathbf{X}})^T(\mathbf{X} - \bar{\mathbf{X}})\right)^{-1}(\mathbf{X} - \bar{\mathbf{X}})^T(\mathbf{y} - \bar{\vec{y}}) \tag{7.41}$$

$$w_0 = \mu_Y - \mathbf{w}^T\boldsymbol{\mu}_X = \overline{y} - \mathbf{w}^T\overline{\mathbf{x}} \tag{7.42}$$

2. The generative model can handle missing inputs, and can be used for semi-supervised learning ($\mathbf{x}$ is known but y is not). However, the generative model assumes $\mathbf{X}, Y$ is jointly Gaussian, which is unlikely to be true for many kinds of features, especially discrete inputs.

7.1.10 Bayesian linear regression using the g-prior

Not done yet.

Chapter 8

Logistic regression

8.1 Solutions

8.1.1 Spam classification using logistic regression

Listing 8.1: emailClassifyLogreg.m

```
% use the spam email data from Hastie book 2e p301


nLambda = 5;
lambdas = logspace(-5,1,nLambda);
Nfolds = 5;

methods = {'stnd', 'log', 'binary'};
fprintf('%s \t %s \t%s \n', 'method', 'trainErr', 'testErr');

for m=1:length(methods)
  load spamData
  method = methods{m};
  switch method
    case 'stnd'
      [Xtrain, mu, sig] = standardizeCols(Xtrain);
      Xtest = standardizeCols(Xtest, mu, sig);
    case 'log'
      Xtrain = log(Xtrain + 0.1);
      Xtest = log(Xtest + 0.1);
    case 'binary'
      Xtrain = (Xtrain>0);
      Xtest = (Xtest > 0);
  end

  [model, Lstar, mu, se] = fitCv(lambdas, ...
    @logregL2Fit, @logregPredict, @zeroOneLossFn, ...
    Xtrain, ytrain,  Nfolds);

  figure;
  ndx = log(lambdas);
  errorbar(ndx, mu, se);
  xlabel('log(lambda)')
  ylabel('CV misclassification rate')
  best = find(Lstar==lambdas);
  title(method)
  %h = line(ndx([best best]), [min(mu) max(mu)]);

  yhatTrain = logregPredict(model, Xtrain);
  errRateTrain = mean( (yhatTrain ~= ytrain) );

  yhatTest = logregPredict(model, Xtest);
  errRateTest = mean( (yhatTest ~= ytest) );

  fprintf('%s \t %5.3f \t %5.3f\n', ...
    method, errRateTrain, errRateTest);
end
```

8.1.2 Spam classification using naive Bayes

These are the results I get with the code below.

```
method   train test
binary   0.108 0.109
```

```
stdn      0.177 0.187
log       0.164 0.182
```

Here is the code, written by Farbod Roosta-Khorasani.

Listing 8.2: naiveBayesGaussSpamDemo.m

```
function naiveBayesGaussSpamDemo()
load('spamData.mat')
Y_train = ytrain + 1;
Y_test = ytest + 1;

%% Binary features
X_train01 = Xtrain > 0;
X_test01 = Xtest > 0;

model = naiveBayesBerFit(X_train01, Y_train,1);
yhat = naiveBayesBerPredict(model, X_train01);
binary_MSE_Train = mean(zeroOneLossFn(Y_train, yhat));
yhat = naiveBayesBerPredict(model, X_test01);
binary_MSE_Test = mean(zeroOneLossFn(Y_test, yhat));

format compact
fprintf('%-8s\t%s\t%s\n', 'method', 'train', 'test')
fprintf('%-8s\t%.3f\t%.3f\n', 'binary', binary_MSE_Train, binary_MSE_Test);


%% Gaussian model on standardized features

%X_train = zscore(Xtrain);
%X_test = zscore(Xtest);

X_train = standardize(Xtrain);
X_test = standardize(Xtest);

model = naiveBayesGaussianFit(X_train, Y_train);

yhat = naiveBayesGaussianPredict(model, X_train);
stdn_MSE_Train = mean(zeroOneLossFn(Y_train,yhat));

yhat = naiveBayesGaussianPredict(model, X_test);
stdn_MSE_Test = mean(zeroOneLossFn(Y_test,yhat));

fprintf('%-8s\t%.3f\t%.3f\n', 'stdn', stdn_MSE_Train, stdn_MSE_Test);


%% Gaussian model on log features

X_train = log(Xtrain+0.1);
X_test = log(Xtest+0.1);

model = naiveBayesGaussianFit(X_train, Y_train);

yhat = naiveBayesGaussianPredict(model, X_train);
log_MSE_Train = mean(zeroOneLossFn(Y_train,yhat));

yhat = naiveBayesGaussianPredict(model, X_test);
log_MSE_Test = mean(zeroOneLossFn(Y_test,yhat));

fprintf('%-8s\t%.3f\t%.3f\n', 'log', log_MSE_Train, log_MSE_Test);

end

%%%


function model = naiveBayesGaussianFit(Xtrain, ytrain)

C = length(unique(ytrain));
[Ntrain, D] = size(Xtrain);
for c=1:C
    index = find(ytrain==c);
    xtr = Xtrain(index,:);
    mu(c,:) = mean(xtr);
    sigma(c,:) = std(xtr,1); % use MLE
    Nclass(c) = length(index);
end
model.classPrior = normalize(Nclass);
model.mu = mu;
model.sigma = sigma;
end

function [yhat, py] = naiveBayesGaussianPredict(model, Xtest)

classPrior = model.classPrior;
computeProb = (nargout >= 2);
[Ntest, D] = size(Xtest);
C = length(classPrior);
```

```
if nargin < 3, classPrior = (1/C)*ones(1,C);
end
logPrior = log(classPrior);
loglik = zeros(1,C);

yhat = zeros(Ntest,1);
py = zeros(Ntest,1);
for i = 1:Ntest
    for c=1:C
        muC = model.mu(c,:);
        sigma2C = model.sigma(c,:).^2;
        xtr = Xtest(i,:);
        loglik(c) = sum(-0.5*log(2*pi*sigma2C)-0.5*((xtr-muC).^2)./sigma2C);
    end
    logPost = loglik + logPrior;
    yhat(i) = argmax(logPost);
    if computeProb
        py(i) = exp(normalizeLogspace(logPost));
    end
end
end
```

8.1.3 Gradient and Hessian of log-likelihood for logistic regression

1. We have

$$\begin{aligned}
\frac{d\sigma(a)}{da} &= \frac{d}{da}(1+e^{-a})^{-1} && (8.1)\\
&= \frac{e^{-a}}{(1+e^{-a})^2} && (8.2)\\
&= \sigma(a)\frac{e^{-a}}{1+e^{-a}} && (8.3)\\
&= \sigma(a)\left(\frac{1+e^{-a}}{1+e^{-a}} - \frac{1}{1+e^{-a}}\right) && (8.4)\\
&= \sigma(a)\frac{e^{-a}}{1+e^{-a}} && (8.5)\\
&= \sigma(a)\frac{1}{1+e^{a}} && (8.6)\\
&= \sigma(a)(1-\sigma(a)) && (8.7)
\end{aligned}$$

2. Let $\eta_i = \mathbf{w}^T\mathbf{x}_i$ and $\mu_i = \sigma(\eta_i)$. Hence by the chain rule and the previous question we have

$$\begin{aligned}
\frac{\partial}{\partial w_j}\mu_i &= \frac{\partial}{\partial w_j}\sigma(\mathbf{w}^T\mathbf{x}_i) && (8.8)\\
&= \frac{\partial}{\partial \eta_i}\sigma(\eta_i)\frac{\partial \eta_i}{\partial w_j} && (8.9)\\
&= \mu_i(1-\mu_i)x_{ij} && (8.10)
\end{aligned}$$

Thus

$$\begin{aligned}
\nabla_{\mathbf{w}}\log\mu_i &= \frac{\mu_i(1-\mu_i)\mathbf{x}_i}{\mu_i} = (1-\mu_i)\mathbf{x}_i && (8.11)\\
\nabla_{\mathbf{w}}\log(1-\mu_i) &= \frac{-\mu_i(1-\mu_i)\mathbf{x}_i}{1-\mu_i} && (8.12)\\
&= -\mu_i\mathbf{x}_i && (8.13)\\
\nabla_{\mathbf{w}}J(\mathbf{w}) &= -\sum_{i=1}^{n}[y_i(1-\mu_i)\mathbf{x}_i - (1-y_i)\mu_i\mathbf{x}_i] && (8.14)\\
&= -\sum_{i=1}^{n}[y_i\mathbf{x}_i - y_i\mathbf{x}_i\mu_i - \mathbf{x}_i\mu_i + y_i\mathbf{x}_i\mu_i)] && (8.15)\\
&= \sum_{i=1}^{n}(\mu_i - y_i)\mathbf{x}_i && (8.16)
\end{aligned}$$

3. We have, for any nonzero vector $\mathbf{v}$,

$$\mathbf{v}^T\mathbf{X}^T\mathbf{S}\mathbf{X}\mathbf{v} = (\mathbf{v}^T\mathbf{X}^T\mathbf{S}^{\frac{1}{2}})(\mathbf{S}^{\frac{1}{2}}\mathbf{X}\mathbf{v}) = ||\mathbf{v}^T\mathbf{X}^T\mathbf{S}^{\frac{1}{2}}||_2^2 > 0 \tag{8.17}$$

Or we can write this out component-wise:

$$\begin{aligned}\mathbf{v}^T(\mathbf{X}^T\mathbf{S}\mathbf{X})\mathbf{v} &= \sum_i (\mathbf{v}^T\mathbf{x}_i)\mu_i(1-\mu_i)(\mathbf{x}_i^T\mathbf{v}) &(8.18)\\ &= \sum_i \mu_i(1-\mu_i)(\mathbf{v}^T\mathbf{x}_i)^2 > 0 &(8.19)\end{aligned}$$

8.1.4 Gradient and Hessian of log-likelihood for multinomial logistic regression

1. Let us drop the i subscript for simplicity. Let $S = \sum_k e^{\eta_k}$ be the denominator of the softmax.

$$\begin{aligned}\frac{\partial \mu_k}{\partial \eta_j} &= \left[\frac{\partial}{\partial \eta_j} e^{\eta_k}\right] S^{-1} + e^{\eta_k} \cdot [\frac{\partial}{\partial \eta_j} S] \cdot -S^{-2} &(8.20)\\ &= (\delta_{ij} e^{\eta_k}) S^{-1} - e^{\eta_k} e^{\eta_j} S^{-2} &(8.21)\\ &= \frac{e^{\eta_k}}{S}\left(\delta_{ij} - \frac{e^{\eta_j}}{S}\right) &(8.22)\\ &= \mu_k(\delta_{kj} - \mu_j) &(8.23)\end{aligned}$$

2. We have

$$\begin{aligned}\nabla_{\mathbf{w}_j}\ell &= \sum_i \sum_k \frac{\partial \ell}{\partial \mu_{ik}} \frac{\partial \mu_{ik}}{\partial \eta_{ij}} \frac{\partial \eta_{ij}}{\partial \mathbf{w}_j} = \sum_i \sum_k \frac{y_{ik}}{\mu_{ik}} \mu_{ik}(\delta_{jk} - \mu_{ij})\mathbf{x}_i &(8.24)\\ &= \sum_i \sum_k y_{ik}(\delta_{jk} - \mu_{ij})\mathbf{x}_i = \sum_i y_{ij}\mathbf{x}_i - \sum_i (\sum_k y_{ik})\mu_{ij}\mathbf{x}_i &(8.25)\\ &= \sum_i (y_{ij} - \mu_{ij})\mathbf{x}_i &(8.26)\end{aligned}$$

3. We consider a single term $\mathbf{x}_i$ in the log likelihood; we can sum over i at the end. Using the Jacobian expression from above, we have

$$\begin{aligned}\nabla_{\mathbf{w}_{c'}}(\nabla_{\mathbf{w}_c}\ell)^T &= \nabla_{\mathbf{w}_{c'}}((y_{ic} - \mu_{ic})\mathbf{x}_i^T) &(8.27)\\ &= -(\nabla_{\mathbf{w}_{c'}}\mu_{ic})\mathbf{x}_i^T &(8.28)\\ &= -(\mu_{ic}(\delta_{c,c'} - \mu_{i,c'})\mathbf{x}_i)\mathbf{x}_i^T &(8.29)\end{aligned}$$

8.1.5 Symmetric version of L2 regularized multinomial logistic regression

Source: Shafiq Joty

At the optimum we have

$$\begin{aligned}\nabla\ell(\mathbf{w}) - 2\lambda\mathbf{w} &= \mathbf{0} &(8.30)\\ \sum_i (\mathbf{y}_i - \boldsymbol{\mu}_i) \otimes \mathbf{x}_i &= \lambda\mathbf{w} &(8.31)\end{aligned}$$

$$\sum_i \begin{pmatrix} (y_{i1} - \mu_{i1}) \begin{pmatrix} x_{i1} \\ \vdots \\ x_{iD} \end{pmatrix} \\ \vdots \\ (y_{iC} - \mu_{iC}) \begin{pmatrix} x_{i1} \\ \vdots \\ x_{iD} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \lambda \begin{pmatrix} w_{i1} \\ \vdots \\ w_{iD} \end{pmatrix} \\ \vdots \\ \lambda \begin{pmatrix} w_{C1} \\ \vdots \\ w_{CD} \end{pmatrix} \end{pmatrix} \tag{8.32}$$

Hence for any j we have

$$\sum_i \sum_c (y_{ic} - \mu_{ic}) x_{ij} = \lambda \sum_c w_{cj} \tag{8.33}$$

$$\sum_i (\sum_c y_{ic} - \sum_c \mu_{ic}) x_{ij} = \lambda \sum_c w_{cj} \tag{8.34}$$

$$0 = \lambda \sum_c w_{cj} \tag{8.35}$$

and hence $\sum_c w_{cj} = 0$ (since $\lambda > 0$).

Since we don't regularize the w_{c0} terms, to ensure identifiability we need to impose the constraint that $w_{c0} = 0$ for some chosen class c (e.g., $c = C$).

8.1.6 Elementary properties of ℓ_2 regularized logistic regression

1. False. $J(\mathbf{w})$ is convex.
2. False. The L2 penalty does not promote sparsity.
3. True. To maximize the probability, we want the sigmoid function to become a step function, so some weights become infinite.
4. False. As we increase λ, we reduce the flexibility of the model, and hence decrease the logprob of the training data.
5. False. As we change λ from 0 to infinitiy, we initially overfit and then underfit. So the logprob of the testset vs λ has a U-shape.

8.1.7 Regularizing separate terms in 2d logistic regression

1. Unregularized solution: see Figure 8.1(a). There is a unique ML decision boundary which makes 0 errors, but there are several possible decision boundaries which all makes **0 errors**.
2. Heavily regularizing w_0 sets $w_0 = 0$, so the line must go through the origin. There are several possible lines with different slopes, but all will make **1 error**. see Figure 8.1(b).
3. Regularizing w_1 makes the line horizontal since x_1 is ignored. This incurs **2 errors**. see Figure 8.1(c).
4. Regularizing w_2 makes the line vertical, since x_2 is ignored. This incurs **0 errors**. see Figure 8.1(d).

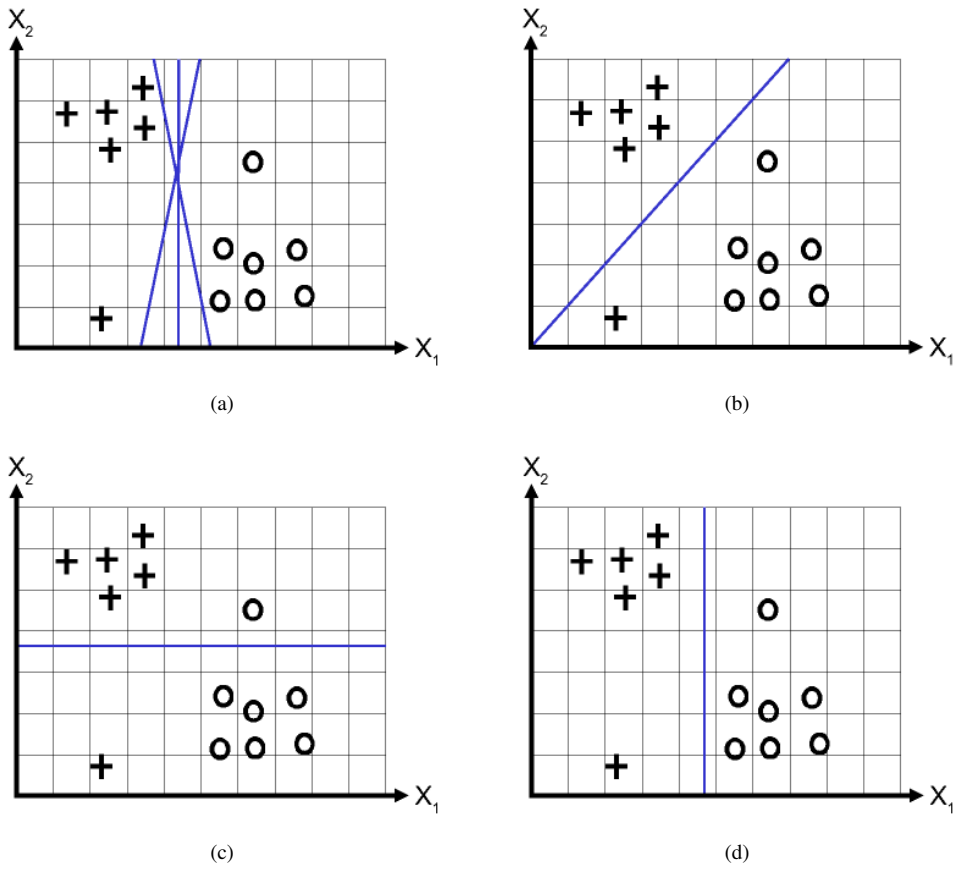


Figure 8.1: (a) Unregularized. (b) Regularizing w_0. (c) Regularizing w_1. (d) Regularizing w_2.

Chapter 9

Generalized linear models and the exponential family

9.1 Solutions

9.1.1 Conjugate prior for univariate Gaussian in exponential family form

Source: (Bernardo and Smith 1994)

The normal likelihood with unknown mean μ and precision λ is given by

$$p(\mathcal{D}|\mu,\lambda) = \left(\prod_{i=1}^{N}\frac{\lambda}{2\pi}\right)^{\frac{1}{2}}\exp(-\frac{\lambda}{2}(x_i-\mu)^2) \tag{9.1}$$

$$= (2\pi)^{-N/2}\left[\lambda^{\frac{1}{2}}\exp(-\frac{\lambda}{2}\mu^2)\right]^N \exp\left[\mu\lambda\sum_i x_i - \frac{\lambda}{2}\sum_i x_i^2\right] \tag{9.2}$$

Hence the conjugate prior has the form

$$p(\mu,\lambda|\nu_0,\tau_1,\tau_2) \propto \left[\lambda^{\frac{1}{2}}\exp(-\frac{\lambda}{2}\mu^2)\right]^{\nu_0}\exp\left[\mu\lambda\tau_1 - \frac{\lambda}{2}\tau_2\right] \tag{9.3}$$

Writing $\alpha = \frac{1}{2}(\nu_0+1)$, $\beta = \frac{1}{2}(\tau_2 - \frac{\tau_1^2}{\nu_0})$, and $\gamma = \frac{\tau_1}{\tau_0}$ (for $\alpha > \frac{1}{2}$, $\beta > 0$ and $\gamma \in \mathbb{R}$) we see that this is a normal-gamma density:

$$p(\mu,\lambda|\nu_0,\tau_1,\tau_2) = \mathcal{N}(\mu|\gamma,\lambda(2\alpha-1))\mathrm{Ga}(\lambda|\alpha,\beta) \tag{9.4}$$

Notice how the conjugate prior only has three parameters, whereas the semi-conjugate prior (Section **??**) has four.

9.1.2 The MVN is in the exponential family

We have

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \underbrace{(2\pi)^{-d/2}}_{h(\mathbf{x})}\underbrace{\exp\left[\frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Lambda}\boldsymbol{\mu}\right]}_{g(\boldsymbol{\eta})}\exp\left[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Lambda}\mathbf{x} + \mathbf{x}^T\boldsymbol{\Lambda}\boldsymbol{\mu}\right] \tag{9.5}$$

$$= h(\mathbf{x})g(\boldsymbol{\eta})\exp\left[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Lambda}\mathbf{x} + \mathbf{x}^T\boldsymbol{\Lambda}\boldsymbol{\mu}\right] \tag{9.6}$$

$$= h(\mathbf{x})g(\boldsymbol{\eta})\exp\left[-\frac{1}{2}(\sum_{ij}x_i x_j\Lambda_{ij}) + \mathbf{x}^T\boldsymbol{\xi}\right] \tag{9.7}$$

$$= h(\mathbf{x})g(\boldsymbol{\eta})\exp\left[-\frac{1}{2}\mathrm{vec}(\boldsymbol{\Lambda})^T\mathrm{vec}(\mathbf{x}\mathbf{x}^T) + \boldsymbol{\xi}^T\mathbf{x}\right] \tag{9.8}$$

$$= h(\mathbf{x})g(\boldsymbol{\eta})\exp\left[\boldsymbol{\eta}^T\mathbf{T}(\mathbf{x})\right] \tag{9.9}$$

where

$$h(\mathbf{x}) = (2\pi)^{-d/2} \tag{9.10}$$

$$\boldsymbol{\eta} = \begin{pmatrix} \boldsymbol{\xi} & \text{vec}(\boldsymbol{\Lambda}) \end{pmatrix}^T \tag{9.11}$$

$$g(\boldsymbol{\eta}) = \exp\left[\frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Lambda}\boldsymbol{\mu}\right] = \exp\left[\frac{1}{2}\log|\boldsymbol{\Lambda}| - \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{\xi}\right] \tag{9.12}$$

$$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} \mathbf{x} & -\frac{1}{2}\text{vec}(\mathbf{x}\mathbf{x}^T) \end{pmatrix}^T \tag{9.13}$$

and $\text{vec}(\mathbf{A}) = [\mathbf{A}(:,1); \mathbf{A}(:,2); \ldots; \mathbf{A}(:,n)]$ concatenates the columns of a matrix. This is analogous to the univariate result in Equation **??**.

Chapter 10

Directed graphical models (Bayes nets)

10.1 Solutions

10.1.1 Marginalizing a node in a DGM

The original DAG is shown in Figure 10.1(a). One minimal I-map for the distribution without X is shown in Figure 10.1(b). An alternative is if the $E \to F$ arrow is reversed, and the $C \to F$ arrow gets replaced by $D \to E$.

Let us know justify why we added these edges:

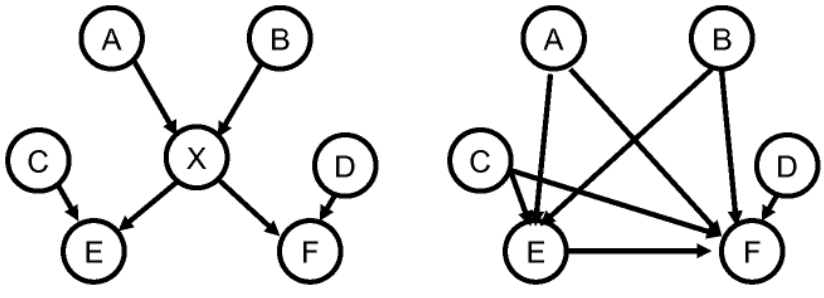
1. $A \to E$. If X is removed, it cannot block the $A \to E$ path, so we must add the edge.
2. $A \to F$. If X is removed, it cannot block the $A \to F$ path.
3. $B \to E$. If X is removed, it cannot block the $B \to E$ path.
4. $B \to F$. If X is removed, it cannot block the $B \to F$ path.
5. $E \to F$ (or $F \to E$). If X is gone, it cannot block the path $E \leftarrow X \to F$ that existed in the original network.
6. $C \to F$ (or $D \to E$). If E is observed, there is an active path $C \to E \to X \to F$ in the original network which can no longer be blocked.

The general rule is: for every direct descendant of X, we add arcs to it from the parents of X (here A, B), from the other direct descendants of X prior in the ordering (here E, which is prior to F), and from the parents of the previously added direct descendants (here C, which is parent of E).

What about the other nodes not in X's family, (which is its parents, children, and co-parents)? They are unaffected. If a node X_m is independent of $X_1, \ldots, X_{m-1}$ (including X) given its parents, it is also independent of $X_1, \ldots, X_{m-1} \setminus \{X\}$ given the (same set of) parents.

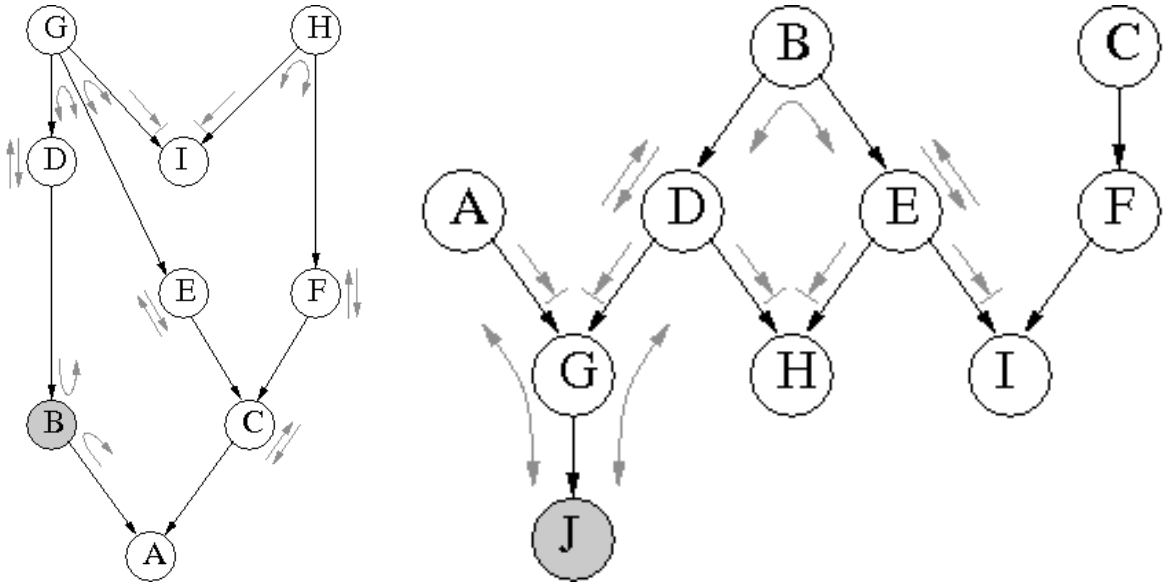
10.1.2 Bayes Ball

1. B does not d-separate A from any of the other nodes, so A is not independent of any of them (given B). See Figure 10.2(a).
2. Starting the ball at A, we can reach B, D, E, G, H and I. So $A \perp \{C, F\}|J$. See Figure 10.2(b).



(a) (b)

Figure 10.1: (a) A small DGM. (b) With X marginalized out.



(a) (b)

Figure 10.2: Bayes ball solutions

10.1.3 Markov blanket for a DGM

Partition all the nodes into X_i and the other nodes, X_{-i}. We can partition the other nodes X_{-i} into those that involve X_i (namely its parents, its children, and its co-parents), and the other nodes, O. Then the full conditional is given by

$$
\begin{aligned}
p(X_i|X_{-i}) &= \frac{p(X_i, X_{-i})}{\sum_x p(X_i = x, X_{-i})} && (10.1)\\
&= \frac{p(X_i, U_{1:n}, Y_{1:m}, Z_{1:m}, O)}{\sum_x p(X_i = x, U_{1:n}, Y_{1:m}, Z_{1:m}, O)} && (10.2)\\
&= \frac{p(X_i|U_{1:n})[\prod_j p(Y_j|X_i, Z_j)]P(U_{1:n}, Z_{1:m}, O)}{\sum_x p(X_i = x|U_{1:n})[\prod_j p(Y_j|X_i = x, Z_j)]P(U_{1:n}, Z_{1:m}, O)} && (10.3)\\
&= \frac{p(X_i|U_{1:n})[\prod_j p(Y_j|X_i, Z_j)]}{\sum_x p(X_i = x|U_{1:n})[\prod_j p(Y_j|X_i = x, Z_j)]} && (10.4)\\
&\propto p(X_i|Pa(X_i)) \prod_{Y_j \in ch(X_i)} p(Y_j|Pa(Y_j)) && (10.5)
\end{aligned}
$$

so the terms that do not involve X_i cancel out from the numerator and denominator. We are left with a product of terms that include X_i in their "scope". This proves that $X_i \perp X_{R_i}|MB_i$.

10.1.4 Hidden variables in DGMs

1. For the graph on the left, the CPDs for nodes 1,2,3 have 1 free parameter each (since they are Bernoulli). $p(H|X_{1:3})$ has 8 free parameters, one per conditioning case. $p(X_i|H)$ for $i = 4 : 6$ are 2×2 tables, but due to the sum to one constraint, only have 2 free parameters. Hence in total there are $3 \times 1 + 8 + 3 \times 2 = 17$.

2. For the graph on the right, the CPDs for nodes 1,2,3 have 1 free parameter each (since they are Bernoulli). $p(X_4|X_{1:3})$ has 8 parameters, one per conditioning case. $p(X_5|X_{1:4})$ has 16 parameters. $p(X_6|X_{1:5})$ has 32 parameters. In total there are $3 + 8 + 16 + 32 = 59$ parameters.

3. The model on the left has a hidden variable, so the exact likelihood is multimodal. We can use EM to find an a local optimum. (Since H only has 2 states, we can optimize $p(X_{1:6}|H, \theta)$ for each value of H separately, but in general this is intractable.) The model on the right is fully observed, so we can easily find the MLE, but it has more parameters, and hence needs more data.

10.1.5 Bayes nets for a rainy day

1. The expression for $P(S=1|V=1)$ is

$$\begin{aligned} P(S=1|V=1) &= \frac{P(S=1,V=1)}{P(V=1)} \\ &= \frac{1}{P(V=1)} \sum_{R=0}^{1} \sum_{G=0}^{1} P(V=1)P(G)P(R|V=1,G)P(S=1|G) \\ &= \sum_{RG} P(G)P(R|V=1,G)P(S=1|G) \\ &= \sum_{G} P(G)P(S=1|G) \sum_{R} P(R|V=1,G) \\ &= \sum_{G} P(G)P(S=1|G) \quad (**) \\ &= P(G=0)P(S=1|G=0) + P(G=1)P(S=1,G=1) \\ &= \alpha(1-\gamma) + (1-\alpha)(1-\beta) \\ &= \alpha\gamma + 1 - \beta + \alpha\beta \end{aligned}$$

   The line marked ** follows since $\sum_R P(R|\cdot) = 1$, since CPTs sum to one for every conditioning context.

2. We find $P(S=1|V=0) = P(S=1|V=1)$, since the above expression is independent of V. We can also see this from the graph: by d-separation, $S \perp V$, since R is a hidden child that blocks the ball.

3. MLEs can be estimated by counting events. Thus $\hat{\alpha} = \frac{\#(G=0)}{\#(G=*)} = 1/3$, where $\#(G=0)$ is the number of times G took value 0, and $\#(G=*)$ is the number of times G took on any value. Similarly, $\hat{\beta} = \frac{\#(G=1,S=0)}{\#(S=0)} = 0/1 = 0$ and $\hat{\gamma} = \frac{\#(G=0,S=0)}{\#(S=0)} = 1/1 = 1$.

10.1.6 Fishing nets

1. We have

$$\begin{aligned} p(x_2, x_4 = t) &= \sum_{x1} \sum_{x3} p(x_1, x_2, x_3, x_4 = t) && (10.6) \\ &= \sum_{x1} \sum_{x3} p(x_3|x_2)p(x_4 = t|x_2)p(x_2|x_1)p(x_1) && (10.7) \\ &= p(x_4 = t|x_2)[\sum_{x1} p(x_2|x_1)p(x_1)][\sum_{x3} p(x_3|x_2)] && (10.8) \\ &= \begin{pmatrix} 0.6 \\ 0.05 \end{pmatrix} \begin{pmatrix} 0.5 \times 0.9 + 0.5 \times 0.8 \\ 0.5 \times 0.1 + 0.5 \times 0.2 \end{pmatrix} && (10.9) \\ &= \begin{pmatrix} 0.6 \\ 0.05 \end{pmatrix} \begin{pmatrix} 0.85 \\ 0.15 \end{pmatrix} && (10.10) \\ &= \begin{pmatrix} 0.51 \\ 0.0075 \end{pmatrix} && (10.11) \end{aligned}$$

   (Note that X_3 is a barren node for this query, since $\sum_{x_3} p(x_3|x_2) = 1$.) Hence

$$p(x_4 = t) = 0.51 + 0.0075 = 0.5175 \quad (10.12)$$

   so

$$p(x_2|x_4 = t) = \frac{1}{0.5175} \begin{pmatrix} 0.51 \\ 0.0075 \end{pmatrix} = \begin{pmatrix} 0.9875 \\ 0.0145 \end{pmatrix} \quad (10.13)$$

   so the fish is much more likely to be salmon.

2. We have

$$p(x_1|x_3 = \text{medium}, x_4 = \text{thin}) = \frac{p(x_1, x_3 = m, x_4 = t)}{p(x_3 = m, x_4 = t)} \quad (10.14)$$

The numerator is given by

$$p(x_1, x_3 = m, x_4 = t) = \sum_{x_2} p(x_4 = t|x_2)p(x_3 = m|x_2)p(x_2|x_1)p(x_1) \tag{10.15}$$

$$= p(x_1) \begin{pmatrix} 0.9 \times 0.6 \times 0.33 + 0.1 \times 0.05 \times 0.1 \\ 0.3 \times 0.6 \times 0.33 + 0.7 \times 0.05 \times 0.1 \\ 0.4 \times 0.6 \times 0.33 + 0.6 \times 0.05 \times 0.1 \\ 0.8 \times 0.6 \times 0.33 + 0.2 \times 0.05 \times 0.1 \end{pmatrix} \tag{10.16}$$

$$= p(x_1) \begin{pmatrix} 0.1787 \\ 0.0629 \\ 0.0822 \\ 0.1594 \end{pmatrix} \tag{10.17}$$

So

$$p(x_1|x_3 = \text{medium}, x_4 = \text{thin}) = \begin{pmatrix} 0.3698 \\ 0.1302 \\ 0.1701 \\ 0.3299 \end{pmatrix} \tag{10.18}$$

so winter is the most likely season.

10.1.7 Removing leaves from BN20

- We have

$$p(z_{1:3}|x_1, x_2, x_4) \propto \sum_{x_3} \sum_{x_5} p(z_{1:3}, x_{1:5}) \tag{10.19}$$

$$= p(z_{1:3})p(x_1|z_{1:3})p(x_2|z_{1:3})p(x_4|z_{1:3}) \left[\sum_{x_3} p(x_3|z_{1:3})\right] \left[\sum_{x_5} p(x_5|z_{1:3})\right] \tag{10.20}$$

$$= p(z_{1:3})p(x_1|z_{1:3})p(x_2|z_{1:3})p(x_4|z_{1:3}) \tag{10.21}$$

since $\sum_{x_3} p(x_3|z_{1:3}) = 1.$ and $\sum_{x_5} p(x_3|z_{1:3}) = 1$. Note that we cannot remove hidden "leaves" from a UGM, since potentials do not necessarily sum to one locally.

- See (Jaakkola and Jordan 1999) for the details.

10.1.8 Handling negaitive findings in the QMR network

The key observation is that the prior $p(\mathbf{d})$ factorizes over d_s, and so does the likelihood, since $p(f_t = 0|\mathbf{d}) = \prod_s p(f_t = 0|d_s)$. Hence the posterior also factorizes into a product of terms, one per disease. For example, suppose we have 2 negative findings and 2 diseases. Then

$$p(\mathbf{d}|\mathbf{f}^-) \propto p(d_1, d_2, f_1, f_2) = p(d_1)p(d_2)p(f_1|d_1)p(f_1|d_2)p(f_2|d_1)p(f_2|d_2) \tag{10.22}$$

$$= [p(d_1)p(f_1|d_1)p(f_2|d_1)][p(d_2)p(f_1|d_2)p(f_2|d_2)] \tag{10.23}$$

10.1.9 Moralization does not introduce new independence statements

Consider a CI statement in M of the form S separates A from B. Consider nodes $s \in S$, $a \in A$, $b \in B$. Suppose in the DAG we have a v-structure, $a \to s \leftarrow b$. (We say the arrows meet head-to-head.) This would not imply that $a \perp b|s$ in the DAG, because of explaining away. However, moralization will always add an edge between a and b, so this situation cannot occur. (A v-structure where the parents are not connected is called a unshielded collider. Moralization eliminates such cases.) Therefore we must have one of these 3 structures: $a \leftarrow s \to b$, $a \to s \to b$, or $a \leftarrow s \leftarrow b$. In all 3 cases, we have $a \perp b|s$ in the DAG as well as the UGM. Hence the UGM cannot make any CI claims that are not part of the DAG.

Of course, the opposite is not true: the act of moralization loses some of the CI statements of the DAG. But it is okay if the graph does not make any CI claims, as long as it does not make any false CI claims. If the original DAG has no v-structures, then moralization does not lose any information. This can only be the case if the original DAG was a simple tree (not a polytree).

Chapter 11

Mixture models and the EM algorithm

11.1 Solutions

11.1.1 T distribution is an infinite sum of Gaussians

To see this, note that

$$p(x) = \int_0^\infty \mathcal{N}(x|0,1/\lambda)\mathrm{Ga}(\lambda|\nu/2,\nu/2)d\lambda \tag{11.1}$$

$$= \int (2\pi)^{-\frac{1}{2}}\lambda^{\frac{1}{2}}\exp(-\frac{\lambda}{2}x^2)\times(\nu/2)^{\nu/2}\Gamma(\nu/2)^{-1}\lambda^{\frac{\nu}{2}-1}\exp(-\frac{\lambda\nu}{2})d\lambda \tag{11.2}$$

$$= c\int \lambda^{\frac{\nu-1}{2}}\exp(-\frac{\lambda}{2}(x^2+\nu))d\lambda \tag{11.3}$$

where we defined $c = (2\pi)^{-\frac{1}{2}}(\nu/2)^{\nu/2}\Gamma(\nu/2)^{-1}$. Now let $\Delta = (x^2+\nu)/2$ and $z = \lambda\Delta$. Then

$$p(x) = c\int z^{\frac{\nu-1}{2}}\Delta^{-1}e^{-z}dz \tag{11.4}$$

$$= c\Gamma(\frac{\nu+1}{2})(\frac{x^2+\nu}{2})^{-\frac{\nu+1}{2}} \tag{11.5}$$

where we used the definition of the Gamma function (Equation **??**). Finally, collecting together all the constants, we have

$$p(x) = (2\pi)^{-\frac{1}{2}}(\nu/2)^{\nu/2}\Gamma(\nu/2)^{-1}\Gamma(\frac{\nu+1}{2})\left[\frac{\nu}{2}(\frac{x^2}{\nu}+1)\right]^{-\frac{\nu+1}{2}} \tag{11.6}$$

$$= (\frac{2\pi\nu}{2})^{-\frac{1}{2}}\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\nu/2)}(1+\frac{x^2}{\nu})^{-\frac{\nu+1}{2}} = \mathcal{T}(x|0,1,\nu) \tag{11.7}$$

Alternatively, we can write

$$\mathcal{T}(x|0,1,\nu) = \int_0^\infty \mathcal{N}(x|0,\tau^2)\mathrm{IG}(\tau^2|\nu/2,\nu/2)d\tau \tag{11.8}$$

$$\frac{b^a}{\Gamma(a)}\left(\frac{1}{2\pi}\right)^{\frac{1}{2}}\Delta^{-a-\frac{1}{2}}\Gamma(a+\frac{1}{2}) = \frac{\Gamma(a+1/2)}{\Gamma(a)}b^a\left(\frac{1}{2\pi}\right)^{\frac{1}{2}}\Delta^{-a-\frac{1}{2}} \tag{11.9}$$

$$= \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)}\left(\frac{\nu}{2\lambda}\right)^{\nu/2}\left(\frac{1}{2\pi}\right)^{\frac{1}{2}}\left(\frac{\nu}{2\lambda}+\frac{(x-\mu)^2}{2}\right)^{-(\nu+1)/2} \tag{11.10}$$

$$= \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)}\left(\frac{\nu}{2\lambda}\right)^{\nu/2}\left(\frac{1}{2\pi}\right)^{\frac{1}{2}}\left(\frac{\nu}{2\lambda}\left[1+\frac{\lambda}{\nu}(x-\mu)^2\right]\right)^{-(\nu+1)/2} \tag{11.11}$$

$$= \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)}\left(\frac{\lambda}{\nu\pi}\right)^{\frac{1}{2}}\left[1+\frac{\lambda}{\nu}(x-\mu)^2\right]^{-(\nu+1)/2} \tag{11.12}$$

$$= \mathcal{T}_\nu(x|\mu,\lambda^{-1}) \tag{11.13}$$

11.1.2 EM for mixtures of Gaussians

Using the following identities,

$$\frac{\partial \mathbf{x}^T \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x} \tag{11.14}$$

$$\frac{\partial \log |\mathbf{X}|}{\partial \mathbf{X}} = (\mathbf{X}^{-1})^T = (\mathbf{X}^T)^{-1} \tag{11.15}$$

$$\log |\mathbf{X}| = -\log |\mathbf{X}^{-1}| \tag{11.16}$$

$$\frac{\partial \mathbf{a}^T \mathbf{X}\mathbf{b}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{b}^T \tag{11.17}$$

for the mean, we have

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k} - \frac{1}{2}\sum_i \sum_k r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k) \tag{11.18}$$

$$= -\frac{1}{2}\sum_i r_{ik}(-2\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)) = 0 \tag{11.19}$$

so

$$\sum_i r_{ik}\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_i = \sum_i r_{ik}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k \tag{11.20}$$

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}} \tag{11.21}$$

For the covariance we have

$$0 = \frac{\partial}{\partial \boldsymbol{\Lambda}_k}\frac{1}{2}\sum_i \sum_k r_{ik} \log |\boldsymbol{\Lambda}_k| - \frac{\partial}{\partial \boldsymbol{\Lambda}}\frac{1}{2}\sum_i \sum_k r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Lambda}_k(\mathbf{x}_i - \boldsymbol{\mu}_k) \tag{11.22}$$

$$= \frac{1}{2}(\sum_i r_{ik})\boldsymbol{\Lambda}_k^{-1} - \frac{1}{2}\sum_i r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \tag{11.23}$$

Hence

Taking derivatives in the usual way results in the following

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}} \tag{11.24}$$

$$\boldsymbol{\Sigma}_k^{new} = \frac{\sum_i r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k^{new})(\mathbf{x}_i - \boldsymbol{\mu}_k^{new})^T}{\sum_i r_{ik}} = \frac{\sum_i r_{ik}\mathbf{x}_i\mathbf{x}_i^T - r_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^T}{r_k} \tag{11.25}$$

11.1.3 EM for mixtures of Bernoullis

- The M step for the μ_{kj} terms is derived as follows. First we compute the gradient:

$$\frac{\partial Q}{\partial \mu_{kj}} = \sum_i r_{ik}\left[\frac{x_{ij}}{\mu_{kj}} - \frac{1 - x_{ij}}{1 - \mu_{kj}}\right] \tag{11.26}$$

$$= \sum_i r_{ik}\frac{(1 - \mu_{kj})x_{ij} - \mu_{kj}(1 - x_{ij})}{\mu_{kj}(1 - \mu_{kj})} \tag{11.27}$$

$$= \frac{[\sum_i r_{ik}(x_{ij} - \mu_{kj})]}{\mu_{kj}(1 - \mu_{kj})} \tag{11.28}$$

Setting $\frac{\partial Q}{\partial \mu_{kj}} = 0$ gives

$$0 = (\sum_i r_{ik}x_{ij}) - (\sum_i r_{ik})\mu_{kj} \tag{11.29}$$

$$\mu_{kj} = \frac{\sum_i r_{ik}x_{ij}}{\sum_i r_{ik}} \tag{11.30}$$

In other words, we just set $\boldsymbol{\mu}_k$ to a weighted average of all the bit vectors assigned to component k.

- Let us put a conjugate $\text{Beta}(\alpha, \beta)$ prior on each μ_{kj} parameter. (Setting $\alpha = \beta = 1$ gives the ML estimate.) The auxiliary function becomes

$$Q(\theta, \theta^t) = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\left[\log \pi_k + \sum_j x_{ij} \log \mu_{kj} + (1 - x_{ij}) \log(1 - \mu_{kj})\right] \tag{11.31}$$

$$+ \sum_{k=1}^{K}\sum_{j=1}^{D}\left[(\alpha - 1) \log \mu_{kj} + (\beta - 1) \log(1 - \mu_{kj})\right] \tag{11.32}$$

This can be optimized wrt $\boldsymbol{\pi}$ and each $\boldsymbol{\mu}_k$ separately. We have

$$\frac{\partial Q}{\partial \mu_{kj}} = \sum_i r_{ik}\left[\frac{x_{ij}}{\mu_{kj}} - \frac{1 - x_{ij}}{1 - \mu_{kj}}\right] + \frac{\alpha - 1}{\mu_{kj}} - \frac{\beta - 1}{1 - \mu_{kj}} \tag{11.33}$$

$$= \sum_i r_{ik}\frac{(1 - \mu_{kj})x_{ij} - \mu_{kj}(1 - x_{ij})}{\mu_{kj}(1 - \mu_{kj})} + \frac{(1 - \mu_{kj})(\alpha - 1) - \mu_{kj}(\beta - 1)}{\mu_{kj}(1 - \mu_{kj})} \tag{11.34}$$

$$= \frac{[\sum_i r_{ik}(x_{ij} - \mu_{kj})] + (1 - \mu_{kj})(\alpha - 1) - \mu_{kj}(\beta - 1)}{\mu_{kj}(1 - \mu_{kj})} \tag{11.35}$$

Setting $\frac{\partial Q}{\partial \mu_{kj}} = 0$ gives

$$0 = (\sum_i r_{ik}x_{ij}) - (\sum_i r_{ik})\mu_{kj} + (1 - \mu_{kj})(\alpha - 1) - \mu_{kj}(\beta - 1) \tag{11.36}$$

$$\mu_{kj} = \frac{(\sum_i r_{ik}x_{ij}) + \alpha - 1}{(\sum_i r_{ik}) + \alpha + \beta - 2} \tag{11.37}$$

If we use a uniform prior, this reduces to the MLE.

11.1.4 EM for mixture of Student distributions

We now have two latent variables per data point: $\mathbf{z}_i$, which is the 1-of-K encoding specifying which cluster data point i belongs to, and $u_i \sim \text{Ga}(\nu_k/2, \nu_k/2)$, which is the latent scale factor for data point i. (Intuitively small u_i means that data point i is likely to be an outlier, so it will have less influence on the parameter estimates.) We give the details (based on (Lo 2009, p57)) below.

The complete data log likelihood is

$$\ell_c(\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\left[\log[\pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k/u_i)] + \log \text{Ga}(u_i|\nu_k/2, \nu_k/2)\right] \tag{11.38}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} z_{ik}\left[\log \pi_k - \frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{u_i}{2}\delta_{ik} + \frac{\nu_k}{2}\log\frac{\nu_k}{2} - \log\Gamma(\frac{\nu_k}{2})\right. \tag{11.39}$$

$$\left. + \frac{\nu_k}{2}(\log u_i - u_i) + (\frac{D}{2} - 1)\log u_i\right] \tag{11.40}$$

where we have defined the Mahalanobis distance to be

$$\delta_{ik} = (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k) \tag{11.41}$$

To compute the expected complete data log-likelihood $Q(\boldsymbol{\theta}) = \mathbb{E}[\ell_c(\boldsymbol{\theta})]$, we define $\tilde{z}_{ik} = \mathbb{E}[z_{ik}|\mathbf{x}_i, \boldsymbol{\theta}] = p(z_i = k|\mathbf{x}_i, \boldsymbol{\theta})$, $\tilde{u}_{ik} = \mathbb{E}[u_i|\mathbf{x}_i, z_{ik} = 1, \boldsymbol{\theta}]$, and $\tilde{s}_{ik} = \mathbb{E}[\log u_i|\mathbf{x}_i, z_{ik} = 1, \boldsymbol{\theta}]$. These can be computed as follows (this constitutes the E step):

$$\tilde{z}_{ik} = \frac{\pi_k \mathcal{T}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{T}(\mathbf{x}_i|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}, \nu_{k'})} \tag{11.42}$$

$$\tilde{u}_{ik} = \frac{\nu_k + D}{\nu_k + \delta_{ik}} \tag{11.43}$$

$$\tilde{s}_{ik} = \log \tilde{u}_{ik} + \Psi(\frac{\nu_k + D}{2}) - \log(\frac{\nu_k + D}{2}) \tag{11.44}$$

Given these quantities, we can compute the M step as follows:

$$\pi_k = \frac{r_k}{N} \tag{11.45}$$

$$r_k \triangleq \sum_{i=1}^{N} \tilde{z}_{ik} \tag{11.46}$$

$$\boldsymbol{\mu}_k = \frac{\sum_i \tilde{z}_{ik}\tilde{u}_{ik}\mathbf{x}_i}{\sum_i \tilde{z}_{ik}\tilde{u}_{ik}} \tag{11.47}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_i \tilde{z}_{ik}\tilde{u}_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{r_k} \tag{11.48}$$

The M step for ν is not available in closed form. However the derivative of Q wrt ν is given by

$$\frac{dQ}{d\nu_k} = \frac{N_k}{2}\left[\log\frac{\nu}{2} + 1 - \Psi(\frac{\nu}{2})\right] + \frac{1}{2}\sum_{i=1}^{N}\tilde{z}_{ik}(\tilde{s}_{ik} - \tilde{u}_{ik}) \tag{11.49}$$

Alternatively, we can optimize the incomplete data log-likelihood,

$$\sum_i \log(\sum_k \pi_k \mathcal{T}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k)) \tag{11.50}$$

wrt the ν_k, using the most recent values of π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, as in the ECME algorithm. This is usually much faster (and avoids the need to compute $\tilde{s}_{ik}$). See `mixStudentFitEm`, written by Hannes Bretschneider, for an implementation.

Having fit this model, we can identify outliers by looking for small values of $\tilde{u}_{ik}$. This term takes on values in the interval $(0, 1 + D/\nu_k)$, and for moderate ν_k, its mean is about 1. So using a threshold of about 0.5 is reasonable. (One can choose the threshold in a more rigorous way, as described in (Lo 2009, p61).)

11.1.5 Gradient descent for fitting GMM

Not done yet.

11.1.6 EM for a scale mixture of Gaussians

1. The responsibilities are

$$\gamma_{ijk} \triangleq p(j,k|x_i,\theta) = \frac{p(x|j,k,\theta)p(j,k|\theta)}{p(x|\theta)} = \frac{p_j q_k N(x_i;\mu_j,\sigma_k^2)}{\sum_{j'}\sum_{k'} p_{j'}q_{k'}N(x_i;\mu_{j'},\sigma_{k'}^2)}$$

2. The expected complete data log likleihood is

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} p(j,k|x_i,\theta)\log(p_j q_k N(x_i;\mu_j,\sigma_k^2))$$

3. Differentiating wrt μ_j

$$\begin{aligned} 0 &= \frac{\partial}{\partial\mu_j}\sum_{ijk}\gamma_{ijk}\frac{(x_i-\mu_j)^2}{2\sigma_k^2} + \text{ terms independent of } \mu_j \\ &= \sum_{ik}\gamma_{ijk}\frac{2(x_i-\mu_j)}{2\sigma_k^2} \\ \mu_j &= \frac{\sum_{ik}\frac{\gamma_{ijk}x_i}{\sigma_k^2}}{\sum_{ik}\frac{\gamma_{ijk}}{\sigma_k^2}} \end{aligned}$$

11.1.7 Manual calculation of the M step for a GMM

1. Likelihood

$$p(D|\theta) = \prod_{i=1}^{3} \sum_{k=1}^{2} \pi_k \mathcal{N}(x_i|\mu_k, \sigma_k)$$

2. Mixing weights

$$\begin{aligned} \pi_1 &= (1 + 0.4 + 0)/3 = 1.4/3 = 0.46 \\ \pi_2 &= (0 + 0.6 + 1)/3 = 1.6/3 = 0.53 \end{aligned}$$

3. Means

$$\begin{aligned} \mu_1 &= \frac{(1)1 + 0.4(10)}{1.4} = 5/1.4 = 3.57 \\ \mu_2 &= \frac{(0.6)10 + 1(20)}{1.6} = 26/1.6 = 16.25 \end{aligned}$$

11.1.8 Moments of a mixture of Gaussians

1. By the rule of **iterated expectation**

$$\mathbb{E}\left[\mathbb{E}\left[\mathbf{x}|z\right]\right] = \mathbb{E}\left[\mathbf{x}\right] \tag{11.51}$$

we have

$$\mathbb{E}\left[\mathbf{x}\right] = \sum_k p(z=k)\mathbb{E}\left[\mathbf{x}|z=k\right] = \sum_k \pi_k \boldsymbol{\mu}_k \tag{11.52}$$

2. We have

$$\begin{aligned} \mathrm{cov}\left[\mathbf{x}\right] &= \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] - \mathbb{E}\left[\mathbf{x}\right]\mathbb{E}\left[\mathbf{x}\right]^T & (11.53) \\ &= \sum_k p(z=k)\mathbb{E}\left[\mathbf{x}\mathbf{x}^T|z=k\right] - \mathbb{E}\left[\mathbf{x}\right]\mathbb{E}\left[\mathbf{x}\right]^T & (11.54) \end{aligned}$$

We use the fact that

$$\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \mathrm{cov}\left[\mathbf{x}\right] + \mathbb{E}\left[\mathbf{x}\right]\mathbb{E}\left[\mathbf{x}\right]^T \tag{11.55}$$

to get

$$\begin{aligned} \mathrm{cov}\left[\mathbf{x}\right] &= \sum_k \pi_k(\mathrm{cov}\left[\mathbf{x}|z=k\right] + \mathbb{E}\left[\mathbf{x}|z=k\right]\mathbb{E}\left[\mathbf{x}|z=k\right]^T) - \mathbb{E}\left[\mathbf{x}\right]\mathbb{E}\left[\mathbf{x}\right]^T & (11.56) \\ &= \sum_k \pi_k(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^T) - \mathbb{E}\left[\mathbf{x}\right]\mathbb{E}\left[\mathbf{x}\right]^T & (11.57) \end{aligned}$$

11.1.9 K-means clustering by hand

The points are assigned to their closest center. The result is shown in Figure 11.1.

11.1.10 Deriving the K-means cost function

Let us focus on a single cluster k. Setting $\mu = x_{i'}$, we have

$$\begin{aligned} \sum_i (x_i - x_{i'})^2 &= \sum_i (x_i - \overline{x})^2 + n(x_{i'} - \overline{x})^2 & (11.58) \\ \sum_{i'}\sum_i (x_i - x_{i'})^2 &= n\sum_i (x_i - \overline{x})^2 + n\sum_{i'}(x_{i'} - \overline{x})^2 = 2n\sum_i (x_i - \overline{x})^2 & (11.59) \end{aligned}$$

Hence

$$\sum_{k=1}^{K} \sum_{i:z_i=k} \sum_{i':z_{i'}=k} (x_i - x_{i'})^2 = 2\sum_{k=1}^{K} n_k \sum_{C(i)=k} (x_i - \overline{x}_k)^2 \tag{11.60}$$

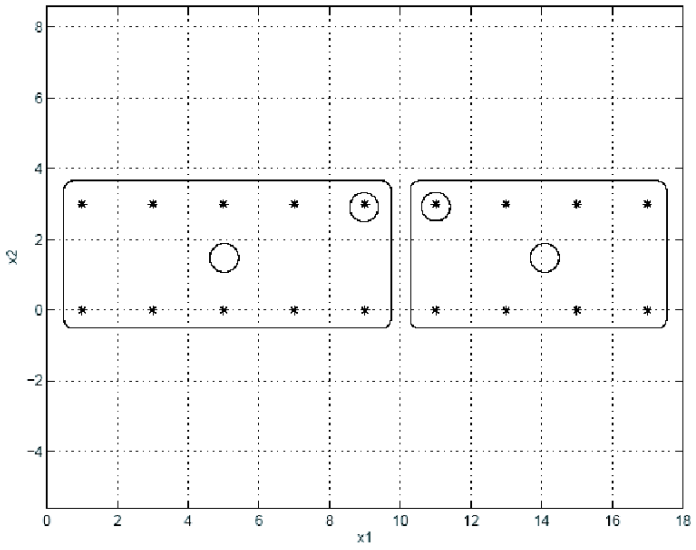


Figure 11.1: Data set for clustering

11.1.11 Visible mixtures of Gaussians are in the exponential family

Source: (Wainwright and Jordan 2008, p46)

First let $z \in \{1, \ldots, K\}$ specify which mixture component the data point comes from, and let $x \in \mathbb{R}$ be the data point. Let $\boldsymbol{\phi}(z) = [\mathbb{I}(z = 1), \ldots, \mathbb{I}(z = K)]$ be the sufficient statistics for z, with canonical parameters $\boldsymbol{\alpha}$. Let $\boldsymbol{\phi}(x) = (x, x^2)$ be the sufficient statistics for x, with canonical parameters $(\gamma_j, \gamma'_j) = (\frac{\mu_k}{\sigma_k^2}, -\frac{1}{2\sigma_k^2})$ for $k = 1 : K$. The overall model has the form

$$p(x, z) = p(z)p(x|z) = \prod_{k=1}^{K} (\alpha_k \mathcal{N}(x|\mu_k, \sigma_k))^{\mathbb{I}(z=k)} \tag{11.61}$$

$$\propto \exp\left\{\sum_k \alpha_k \mathbb{I}(z = k) + \sum_k \mathbb{I}(z = k)[\gamma_k x + \gamma'_k x^2]\right\} \tag{11.62}$$

We can write this in the form $\exp(\boldsymbol{\theta}^T \boldsymbol{\phi}(z, x))$ by defining

$$\boldsymbol{\theta} = [\alpha_1, \ldots, \alpha_K, \gamma_1, \ldots, \gamma_K, \gamma'_1, \ldots, \gamma'_K] \tag{11.63}$$

$$\boldsymbol{\phi}(z, x) = [\mathbb{I}(z = 1), \ldots, \mathbb{I}(z = K), \mathbb{I}(z = 1)x, \ldots, \mathbb{I}(z = K)x, \mathbb{I}(z = 1)x^2, \ldots, \mathbb{I}(z = K)x^2] \tag{11.64}$$

11.1.12 EM for robust linear regression with a Student t likelihood

The complete data log likelihood is given by

$$\log p(\mathbf{y}, \mathbf{z}|\mathbf{X}, \mathbf{w}, \sigma^2, \nu) = \sum_i \left(-\frac{1}{2}\log(2\pi z_i \sigma^2) - \frac{1}{2z_i\sigma^2}(y_i - \mathbf{w}^T \mathbf{x}_i)^2\right. \tag{11.65}$$

$$\left. + \frac{\nu}{2}\log\frac{\nu}{2} + (\frac{\nu}{2} - 1)\log(z_i) - \log\Gamma(\nu/2) - z_i\frac{\nu}{2}\right) \tag{11.66}$$

Ignoring terms not involving $\mathbf{w}$, and taking expectations, we have

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) = \sum_i -\frac{s_i^t}{2\sigma^2}(y_i - \mathbf{w}^T \mathbf{x}_i)^2 \tag{11.67}$$

where

$$s_i^t \triangleq \mathbb{E}\left[1/z_i | y_i, \mathbf{x}_i, \boldsymbol{\theta}^t\right] \tag{11.68}$$

We recognize this as a weighted least squares problem, where s_i^t is the weight of point i at iteration t. (Do not confuse the weights on the data points $\mathbf{s} \in \mathbb{R}^N$ with the model parameters, $\mathbf{w} \in \mathbb{R}^D$.)

We now discuss how to compute these weights. One can show that

$$p(z_i|y_i, \mathbf{x}_i, \boldsymbol{\theta}) \quad = \quad \mathrm{IG}(\frac{\nu+1}{2}, \frac{\nu+\delta_i}{2}) \tag{11.69}$$

where

$$\delta_i = \frac{(y_i - \mathbf{x}^T\mathbf{x}_i)^2}{\sigma^2} \tag{11.70}$$

is the standardized residual. Hence

$$p(1/z_i|y_i, \mathbf{x}_i, \boldsymbol{\theta}) = \mathrm{Ga}(\frac{\nu+1}{2}, \frac{\nu+\delta_i}{2}) \tag{11.71}$$

Furthermore, the mean of a $\mathrm{Ga}(a, b)$ distribution is $\frac{a}{b}$, so

$$\mathbb{E}\left[1/z_i\right] = s_i^t = \frac{\nu^t + 1}{\nu^t + \delta_i^t} \tag{11.72}$$

So if the residual δ_i^t is large, the point will be given low weight, which makes intuitive sense.

11.1.13 EM for EB estimation of Gaussian shrinkage model

E step, we compute $p(\theta_j|\mathcal{D}, \boldsymbol{\eta}^{old})$ using standard Gaussian inference (details are left as an exercise). The M step is very similar to Section **??**, where we fit an MVN to partially observed data, except here we are fitting a scalar Gaussian. We simply use

$$\hat{\mu} = \frac{1}{D}\sum_{j=1}^{D}\mathbb{E}\left[\theta_j\right], \; \hat{\tau}^2 = \frac{1}{D}\left(\sum_{j=1}^{D}\mathbb{E}\left[\theta_j^2\right]\right) - \hat{\mu}^2 \tag{11.73}$$

where $\mathbb{E}\left[\theta_j\right] = \mathbb{E}\left[\theta_j|\mathcal{D}, \boldsymbol{\eta}^{old}\right]$. In the special case that $\sigma_j = \sigma$, this will give the same results as the closed-form solution in the book.

11.1.14 EM for censored linear regression

We follow the presentation of (Tanner 1996, p67). Let us assume a model of the form

$$z_i = \mu_i + \sigma\epsilon_i \tag{11.74}$$

where $\mu_i = w_0 + w_1 x_i$ and $\epsilon_i \sim \mathcal{N}(0, 1)$. Let the first M observations be uncensored, and let observations $M+1$ to N be censored at value c_i. Thus $y_i = z_i$ for $i = 1 : M$ and $y_i = c_i$ for $i = M+1 : N$.

Let z_i be the true but unobserved response for a censored variable. If we knew the z_i's for all the data, the log likelihood would be

$$-\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \sum_{i=1}^{M}\frac{1}{2\sigma^2}(y_i - \mu_i)^2 - \sum_{i=M+1}^{N}\frac{1}{2\sigma^2}(z_i - \mu_i)^2 \tag{11.75}$$

This is called the complete data log likelihood. (Henceforth we will drop the constant term involving $\log(2\pi)$.) Of course, we don't know z_i for $i = M+1 : N$, so let us estimate it. We compute the expected complete data log likelihood as follows

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) \quad = \quad -\frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{M}(y_i - \mu_i)^2 \tag{11.76}$$

$$-\frac{1}{2\sigma^2}\sum_{i=M+1}^{N}\left[\mathbb{E}\left[z_i^2\right] - 2\mu_i\mathbb{E}\left[z_i\right] + \mu_i^2\right] \tag{11.77}$$

where $\boldsymbol{\theta} = (\mathbf{w}, \sigma^2)$ are the unknown parameters, $\boldsymbol{\theta}^t$ are the parameter values at the previous iteration, and the expectations are conditioned on $\boldsymbol{\theta}^t$ and the observed data, i.e., $\mathbb{E}\left[z_i\right] = \mathbb{E}\left[z_i|\boldsymbol{\theta}^t, z_i \geq c_i\right]$ and $\mathbb{E}\left[z_i^2\right] = \mathbb{E}\left[z_i^2|\boldsymbol{\theta}^t, z_i \geq c_i\right]$ for $i = M+1 : N$. (For $i = 1 : M$, we have $\mathbb{E}\left[z_i\right] = y_i$ and $\mathbb{E}\left[z_i^2\right] = y_i^2$.)

For simplicity, let us initially concentrate on estimating $\mathbf{w}$, and treat σ^2 as known In this case, we can cancel terms from Q which don't involve $\mathbf{w}$; hence we can ignore the $\mathbb{E}\left[z_i^2\right]$ term,, which doesn't depend on $\mu_i = \mathbf{w}^T\mathbf{x}_i$, to get the new objective:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^t) \quad = \quad -\frac{1}{2\sigma^2}\sum_{i=1}^{M}(y_i - \mu_i)^2 + \frac{1}{2\sigma^2}\sum_{i=M+1}^{N}\left[2\mu_i\mathbb{E}\left[z_i\right] - \mu_i^2\right] \tag{11.78}$$

From a previous exercise we have that

$$\mathbb{E}\left[z_i|\boldsymbol{\theta}, z_i \geq c_i\right] \quad = \quad \mu_i + \sigma H(\frac{c_i - \mu_i}{\sigma}) \tag{11.79}$$

where we have defined

$$H(u) \triangleq \frac{\phi(u)}{1 - \Phi(u)} \tag{11.80}$$

The M step is then obtained by solving the following equations:

$$\frac{\partial Q}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{i=1}^{M}(y_i - \mu_i) + \sum_{i=M+1}^{N}(\mathbb{E}\left[z_i\right] - \mu_i) = 0 \tag{11.81}$$

$$\frac{\partial Q}{\partial w_1} = 0 \quad \Rightarrow \quad \sum_{i=1}^{M} x_i(y_i - \mu_i) + \sum_{i=M+1}^{N} x_i(\mathbb{E}\left[z_i\right] - \mu_i) = 0 \tag{11.82}$$

In other words, we just do a standard least squares fit, using $\mathbb{E}\left[z_i\right]$ for the response values for the censored cases.

For the M step for σ^2, we have

$$\frac{\partial Q}{\partial \sigma^2} \quad = \quad -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{M}(y_i - \mu_i)^2 + \frac{1}{2\sigma^4}\sum_{i=M+1}^{N}(\mathbb{E}\left[z_i^2\right] - 2\mu_i\mathbb{E}\left[z_i\right] + \mu_i^2) = 0 \tag{11.83}$$

$$\sigma^2 \quad = \quad \frac{1}{N}\left(\sum_{i=1}^{M}(y_i - \mu_i)^2 + \sum_{i=M+1}^{N}(\mathbb{E}\left[z_i^2\right] - 2\mu_i\mathbb{E}\left[z_i\right] + \mu_i^2)\right) \tag{11.84}$$

We can use the expression for $\mathbb{E}\left[z_i^2|z_i \geq c_i\right]$ from a previous exercise.

11.1.15 Posterior mean of a truncated Gaussian

Let us define $\phi(w) = \mathcal{N}(w|0,1)$ and $\Phi(w)$ as the pdf and cdf of the standard Gaussian. We have

$$\mathbb{E}\left[z_i|\boldsymbol{\theta}, z_i \geq c_i\right] \quad = \quad \mu_i + \sigma\mathbb{E}\left[\epsilon_i|\epsilon_i \geq \frac{c_i - \mu_i}{\sigma}\right] \tag{11.85}$$

$$\quad = \quad \mu_i + \sigma\left[\int_{(c_i-\mu_i)/\sigma}^{\infty} w\phi(w)dw\right] / \left[1 - \Phi(\frac{c_i - \mu_i}{\sigma})\right] \tag{11.86}$$

This follows since $p(\epsilon_i|E) = \frac{p(\epsilon_i, E)}{p(E)}$, where E is some event of interest; here the event is $E = \mathbb{I}(z_i \geq c_i) = \mathbb{I}(\epsilon_i \geq \frac{c_i - \mu_i}{\sigma})$. Hence we must renormalize the Gaussian density by dividing by the probability that observation i is censored.

Now, one can show that

$$\frac{d}{dw}\mathcal{N}(w|0,1) = -w\mathcal{N}(w|0,1) \tag{11.87}$$

and hence

$$\int_b^c w\mathcal{N}(w|0,1) = \mathcal{N}(b|0,1) - \mathcal{N}(c|0,1) \tag{11.88}$$

Applying this result to the equation above, we get

$$\mathbb{E}\left[z_i|\boldsymbol{\theta}, z_i \geq c_i\right] \quad = \quad \mu_i + \sigma H(\frac{c_i - \mu_i}{\sigma}) \tag{11.89}$$

where we have defined

$$H(u) \triangleq \frac{\phi(u)}{1 - \Phi(u)} \tag{11.90}$$

For the variance, recall that $z_i = \mu_i + \sigma\epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0,1)$. So $z_i \geq c_i$ is equivalent to $\epsilon_i \geq \frac{c_i - \mu_i}{\sigma}$; call this event E_i. Furthemore, let $a_i = \frac{c_i - \mu_i}{\sigma}$. We have

$$\mathbb{E}\left[z_i^2|E_i\right] \quad = \quad \mathbb{E}\left[\mu_i^2 + 2\mu_i\sigma\epsilon_i + \sigma^2\epsilon_i^2|E_i\right] \tag{11.91}$$

$$\quad = \quad \mu_i^2 + 2\mu_i\sigma\mathbb{E}\left[\epsilon_i|E_i\right] + \sigma^2\mathbb{E}\left[\epsilon_i^2|E_i\right] \tag{11.92}$$

$$\quad = \quad \mu_i^2 + 2\mu_i\sigma H(a_i) + \sigma^2\mathbb{E}\left[\epsilon_i^2|E_i\right] \tag{11.93}$$

Now recall the rule of integration by parts

$$\int u(x)\frac{dv(x)}{dx}dx = u(x)v(x) - \int v(x)\frac{du(x)}{dx}dx \tag{11.94}$$

Let $u(w) = -w$ and $\frac{dv(w)}{dw} = -w\phi(w)$. Then we have

$$\mathbb{E}\left[\epsilon_i^2|E_i\right] = \frac{\int_{a_i}^{\infty} w^2\phi(w)dw}{p(E_i)} \tag{11.95}$$

The integral is given by

$$I_i = \int_{a_i}^{\infty} -w \cdot -w\phi(w)dw = [-w\phi(w)]_{a_i}^{\infty} + \int_{a_i}^{\infty} \phi(w)dw \tag{11.96}$$

$$= a_i\phi(a_i) + 1 - \Phi(a_i) \tag{11.97}$$

So

$$\mathbb{E}\left[\epsilon_i^2|E+i\right] = \frac{1}{1-\Phi(a_i)}[a_i\phi(a_i) + 1 - \Phi(a_i)] = a_iH(a_i) + 1 \tag{11.98}$$

Putting it all together we get

$$\mathbb{E}\left[z_i^2|E_i\right] = \mu_i^2 + 2\mu_i\sigma H(a_i) + \sigma^2 + \sigma(c_i - \mu_i)H(a_i) \tag{11.99}$$

$$= \mu_i^2 + \sigma^2 + \sigma(c_i + \mu_i)H(a_i) \tag{11.100}$$

Chapter 12

Latent linear models

12.1 Solutions

12.1.1 M step for FA

Using the trace trick we have

$$\begin{aligned}
\sum_i \mathbb{E}\left[(\tilde{\mathbf{x}}_i - \mathbf{W}\mathbf{z}_i)^T \boldsymbol{\Psi}^{-1}(\tilde{\mathbf{x}}_i - \mathbf{W}\mathbf{z}_i)\right] &= \sum_i \left[\tilde{\mathbf{x}}_i^T \boldsymbol{\Psi}^{-1}\tilde{\mathbf{x}}_i + \mathbb{E}\left[\mathbf{z}_i^T \mathbf{W}^T \boldsymbol{\Psi}^{-1}\mathbf{W}\mathbf{z}_i\right] - 2\tilde{\mathbf{x}}_i^T \boldsymbol{\Psi}^{-1}\mathbf{W}\mathbb{E}\left[\mathbf{z}_i\right]\right] && (12.1)\\
&= \sum_i \left[\mathrm{tr}(\boldsymbol{\Psi}^{-1}\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^T) + \mathrm{tr}(\boldsymbol{\Psi}^{-1}\mathbf{W}\mathbb{E}\left[\mathbf{z}_i\mathbf{z}_i^T\right]\mathbf{W}^T\right) && (12.2)\\
&\quad -\mathrm{tr}(2\boldsymbol{\Psi}^{-1}\mathbf{W}\mathbb{E}\left[\mathbf{z}_i\right]\tilde{\mathbf{x}}_i^T)\Big] && (12.3)\\
&= \mathrm{tr}(\boldsymbol{\Psi}^{-1}\mathbf{G}(\mathbf{W})) && (12.4)
\end{aligned}$$

Hence the expected complete data log likelihood is given by

$$Q = \frac{N}{2}\log|\boldsymbol{\Psi}^{-1}| - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}^{-1}\mathbf{G}(\mathbf{W})) \quad (12.5)$$

Using the chain rule and the facts that $\frac{\partial}{\partial \mathbf{W}}\mathrm{tr}(\mathbf{W}^T\mathbf{A}) = \mathbf{A}$ $\frac{\partial}{\partial \mathbf{W}}\mathbf{W}^T\mathbf{A}\mathbf{W} = (\mathbf{A} + \mathbf{A}^T)\mathbf{W}$ we have

$$\begin{aligned}
\nabla_{\mathbf{W}} Q(\mathbf{W}) &= -\frac{1}{2}\boldsymbol{\Psi}^{-1}\nabla_{\mathbf{W}} G(\mathbf{W}) = 0 && (12.6)\\
\nabla_{\mathbf{W}} G(\mathbf{W}) &= 2\mathbf{W}\sum_i \mathbb{E}\left[\mathbf{z}_i\mathbf{z}_i^T\right] - 2(\sum_i \mathbb{E}\left[\mathbf{z}_i\right]\tilde{\mathbf{x}}_i^T)^T && (12.7)\\
\mathbf{W}_{mle} &= \left[\sum_i \tilde{\mathbf{x}}_i \mathbb{E}\left[\mathbf{z}_i\right]^T\right]\left[\sum_i \mathbb{E}\left[\mathbf{z}_i\mathbf{z}_i^T\right]\right]^{-1} && (12.8)
\end{aligned}$$

Using the facts that $\nabla_{\mathbf{X}}\log|\mathbf{X}| = \mathbf{X}^{-T}$ and $\nabla_{\mathbf{X}}\mathrm{tr}(\mathbf{X}\mathbf{A}) = \mathbf{A}^T$ we have

$$\begin{aligned}
\nabla_{\boldsymbol{\Psi}^{-1}} Q &= \frac{N}{2}\boldsymbol{\Psi} - \frac{1}{2}\mathbf{G}(\mathbf{W}_{mle}) = 0 && (12.9)\\
\boldsymbol{\Psi} &= \frac{1}{N}\mathrm{diag}(\mathbf{G}(\mathbf{W}_{mle})) && (12.10)
\end{aligned}$$

We can simplify this as follows, by plugging in the MLE (this simplification no longer holds if we use MAP estimation). First note that

$$\sum_i \mathbb{E}\left[\mathbf{z}_i\mathbf{z}_i^T\right]\mathbf{W}_{mle}^T = \sum_i \mathbb{E}\left[\mathbf{z}_i\right]\tilde{\mathbf{x}}_i^T \quad (12.11)$$

so

$$\begin{aligned}
\boldsymbol{\Psi} &= \frac{1}{N}\sum_i \left(\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^T + \mathbf{W}_{mle}\mathbb{E}\left[\mathbf{z}_i\right]\tilde{\mathbf{x}}_i^T - 2\mathbf{W}_{mle}\mathbb{E}\left[\mathbf{z}_i\right]\tilde{\mathbf{x}}_i^T\right) && (12.12)\\
&= \frac{1}{N}\left(\sum_i \tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^T - \mathbf{W}_{mle}\sum_i \mathbb{E}\left[\mathbf{z}_i\right]\tilde{\mathbf{x}}_i^T\right) && (12.13)
\end{aligned}$$

12.1.2 MAP estimation for the FA model

To derive the modified M step, we just need to maximize $\mathcal{Q}'(\boldsymbol{\theta}) = \mathcal{Q}(\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$.

For the regression weights, we have the following, where $\mathbf{P}_i = \mathbb{E}\left[\mathbf{z}_i \mathbf{z}_i^T\right]$:

$$\log p(\mathbf{W}|\boldsymbol{\Psi}) = -\frac{1}{2}\sum_j \mathbf{W}_{j,:}^T (\mathrm{diag}(\boldsymbol{\alpha}) .* \boldsymbol{\Psi}^{-1}) \mathbf{W}_{j,:} \tag{12.14}$$

$$\frac{\partial \mathcal{Q}'}{\partial \mathbf{W}} = +\sum_{i=1}^N \boldsymbol{\Psi}^{-1} \tilde{\mathbf{x}}_i \mathbf{m}_i^T - \sum_{i=1}^N \boldsymbol{\Psi}^{-1} \mathbf{W} \mathbf{P}_i - \boldsymbol{\Psi}^{-1} .* \mathrm{diag}(\boldsymbol{\alpha}) \mathbf{W} = 0 \tag{12.15}$$

$$\hat{\mathbf{W}} = \left(\sum_{i=1}^N \tilde{\mathbf{x}}_i \mathbf{m}_i^T\right) \left(\mathrm{diag}(\boldsymbol{\alpha}) + \sum_{i=1}^N \mathbf{P}_i\right)^{-1} \tag{12.16}$$

For the observation noise, the log prior has the form

$$\log p(\boldsymbol{\Psi}) = \sum_k (a-1) \log \Psi_{kk}^{-1} - \Psi_{kk}^{-1} b \tag{12.17}$$

But we also have a dependence on $\boldsymbol{\Psi}$ from the $p(\mathbf{W}|\boldsymbol{\Psi})$ term. So the objective function has the form

$$\mathcal{Q}' = \frac{N}{2} \log |\boldsymbol{\Psi}^{-1}| - \frac{1}{2} \boldsymbol{\Psi}^{-1} \mathrm{tr}(\mathbf{G}(\hat{\mathbf{W}})) + \sum_k (a-1) \log \Psi_{kk}^{-1} - b \Psi_{kk}^{-1} - \frac{1}{2} \sum_j \sum_k \hat{W}_{jk}^2 \alpha_k \Psi_{jj}^{-1} \tag{12.18}$$

Taking derivatives wrt Ψ_{jj}^{-1} we have

$$\frac{\partial \mathcal{Q}'}{\partial \Psi_{jj}^{-1}} = \frac{N}{2} \Psi_{jj} - \frac{1}{2} G_{jj} + (a-1)\Psi_{jj} - b - \frac{1}{2} \sum_k \hat{W}_{jk}^2 \alpha_k \tag{12.19}$$

Hence the MAP estimate is

$$\Psi_{jj} = \frac{G_{jj} + 2b + \sum_k \hat{W}_{jk}^2 \alpha_k}{2(a-1) + N} \tag{12.20}$$

12.1.3 Heuristic for assessing applicability of PCA

σ^2 measures the spread of the λ_i's. Small σ^2 implies that one may need to use many dimensions in order to capture most of the variance. Large σ^2 means the λ_i's are spread out, so a few dimensions may suffice to capture most of the variance.

12.1.4 Deriving the second principal component

1. Dropping terms that no not involve $\mathbf{z}_2$ we have

$$J = \frac{1}{n} \sum_{i=1}^n \left[-2 z_{i2} \mathbf{v}_2^T (\mathbf{x}_i - z_{i1} \mathbf{v}_1) + z_{i2}^2 \mathbf{v}_2^T \mathbf{v}_2\right] = \frac{1}{n} \sum_{i=1}^n \left[-2 z_{i2} \mathbf{v}_2^T \mathbf{x}_i + z_{i2}^2\right] \tag{12.21}$$

since $\mathbf{v}_2^T \mathbf{v}_2 = 1$ and $\mathbf{v}_1^T \mathbf{v}_2 = 0$. Hence

$$\frac{\partial J}{\partial z_{i2}} = -2 \mathbf{v}_2^T \mathbf{x}_i + 2 z_{i2} = 0 \tag{12.22}$$

so

$$z_{i2} = \mathbf{v}_2^T \mathbf{x}_i \tag{12.23}$$

2. We have

$$\frac{\partial \tilde{J}}{\partial \mathbf{v}_2} = -2 \mathbf{C} \mathbf{v}_2 + 2 \lambda_2 \mathbf{v}_2 + \lambda_{12} \mathbf{v}_1 = 0 \tag{12.24}$$

Premultiplying by $\mathbf{v}_1^T$ yields

$$0 = -2 \mathbf{v}_1^T \mathbf{C} \mathbf{v}_2 + 2 \lambda_2 \mathbf{v}_1^T \mathbf{v}_2 + \lambda_{12} \mathbf{v}_1^T \mathbf{v}_1 \tag{12.25}$$

Now $\mathbf{v}_1^T \mathbf{C} \mathbf{v}_2 = \mathbf{v}_1^T (\lambda_1 \mathbf{v}_2) = 0$, and $\mathbf{v}_1^T \mathbf{v}_2 = 0$, and $\mathbf{v}_1^T \mathbf{v}_1 = 1$, so $\lambda_{12} = 0$. Hence

$$
\begin{aligned}
0 &= -2\mathbf{C}\mathbf{v}_2 + 2\lambda_2 \mathbf{v}_2 && (12.26)\\
\mathbf{C}\mathbf{v}_2 &= \lambda_2 \mathbf{v}_2 && (12.27)
\end{aligned}
$$

So $\mathbf{v}_2$ is an eigenvector of $\mathbf{C}$. Since we want to maximize the variance, we want to pick the eigenvector with the largest eigenvalue, but the first one is already taken. Hence $\mathbf{v}_2$ is the evector with the second largest evalue.

12.1.5 Deriving the residual error for PCA

1. From the previous exercise we have

$$
\begin{aligned}
(\mathbf{x}_i - z_{i1}\mathbf{v}_1 - z_{i2}\mathbf{v}_2)^T (\mathbf{x}_i - z_{i1}\mathbf{v}_1 - z_{i2}\mathbf{v}_2) &= \mathbf{x}_i^T \mathbf{x}_i - 2z_{i1}\cancel{\mathbf{x}_i^T \mathbf{v}_1}^{z_{i1}} - 2z_{i2}\cancel{\mathbf{x}_i^T \mathbf{v}_2}^{z_{i2}} + z_{i1}^2 + z_{i2}^2 && (12.28)\\
&= \mathbf{x}_i^T \mathbf{x}_i - z_{i1}^2 - z_{i2}^2 && (12.29)\\
&= \mathbf{x}_i^T \mathbf{x}_i - \mathbf{v}_1^T \mathbf{x}_i^T \mathbf{x}_i^T \mathbf{v}_1 - \mathbf{v}_2^T \mathbf{x}_i^T \mathbf{x}_i^T \mathbf{v}_2 && (12.30)
\end{aligned}
$$

   This generalizes to $K > 2$ in the obvious way.

2. Hence

$$
\begin{aligned}
J_K &= \frac{1}{n}\sum_{i=1}^n \left(\mathbf{x}_i^T \mathbf{x}_i - \sum_{j=1}^K \mathbf{v}_j^T \mathbf{x}_i \mathbf{x}_i^T \mathbf{v}_j \right) && (12.31)\\
&= \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^T \mathbf{x}_i - \sum_j \mathbf{v}_j^T \left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^T\right) \mathbf{v}_j && (12.32)\\
&= \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^T \mathbf{x}_i - \sum_j \mathbf{v}_j^T \mathbf{C} \mathbf{v}_j && (12.33)\\
&= \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^T \mathbf{x}_i - \sum_{j=1}^K \lambda_j && (12.34)
\end{aligned}
$$

3. If $K = d$ there is no truncation and $J_d = 0$. Hence

$$
0 = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i^T \mathbf{x}_i - \sum_{j=1}^K \lambda_j - \sum_{j=K+1}^d \lambda_j = J_K - \sum_{j=K+1}^d \lambda_j \quad (12.35)
$$

   Hence the residual error arising from only using $K < d$ terms is

$$
J_K = \sum_{j=K+1}^d \lambda_j \quad (12.36)
$$

12.1.6 Derivation of Fisher's linear discriminant

We have

$$
\begin{aligned}
f &= \mathbf{w}^T \mathbf{S}_B \mathbf{w} && (12.37)\\
f' &= 2\mathbf{S}_B \mathbf{w} && (12.38)\\
g &= \mathbf{w}^T \mathbf{S}_W \mathbf{w} && (12.39)\\
g' &= 2\mathbf{S}_W \mathbf{w} && (12.40)\\
\frac{dJ(\mathbf{w})}{d\mathbf{w}} &= \frac{(2\mathbf{s}_B \mathbf{w})(\mathbf{w}^T \mathbf{S}_W \mathbf{w}) - (\mathbf{w}^T \mathbf{S}_B \mathbf{w})(2\mathbf{S}_W \mathbf{w})}{(\mathbf{w}^T \mathbf{S}_W \mathbf{w})^T (\mathbf{w}^T \mathbf{S}_W \mathbf{w})} = 0 && (12.41)
\end{aligned}
$$

Hence

$$
\begin{aligned}
(\mathbf{s}_B \mathbf{w}) \underbrace{(\mathbf{w}^T \mathbf{S}_W \mathbf{w})}_{a} &= \underbrace{(\mathbf{w}^T \mathbf{S}_B \mathbf{w})}_{b} (\mathbf{S}_W \mathbf{w}) && (12.42)\\
a\mathbf{S}_B \mathbf{w} &= b\mathbf{S}_W \mathbf{w} && (12.43)\\
\mathbf{S}_B \mathbf{w} &= \frac{b}{a}\mathbf{S}_W \mathbf{w} && (12.44)
\end{aligned}
$$

12.1.7 PCA via successive deflation

1. We have

$$\begin{aligned}
\tilde{\mathbf{C}} &= \frac{1}{n}\left[(\mathbf{I}-\mathbf{v}_1\mathbf{v}_1^T)\mathbf{X}^T\mathbf{X}(\mathbf{I}-\mathbf{v}_1\mathbf{v}_1^T)\right] && (12.45)\\
&= \frac{1}{n}\left[(\mathbf{X}^T\mathbf{X}-\mathbf{v}_1\mathbf{v}_1^T\mathbf{X}^T\mathbf{X})(\mathbf{I}-\mathbf{v}_1\mathbf{v}_1^T)\right] && (12.46)\\
&= \frac{1}{n}\left[\mathbf{X}^T\mathbf{X}-\mathbf{v}_1(\mathbf{v}_1^T\mathbf{X}^T\mathbf{X})-(\mathbf{X}^T\mathbf{X}\mathbf{v}_1)\mathbf{v}_1^T+\mathbf{v}_1(\mathbf{v}_1^T\mathbf{X}^T\mathbf{X}\mathbf{v}_1)\mathbf{v}_1^T\right] && (12.47)\\
&= \frac{1}{n}\left[\mathbf{X}^T\mathbf{X}-\mathbf{v}_1(n\lambda_1\mathbf{v}_1^T)-(n\lambda_1\mathbf{v}_1)\mathbf{v}_1^T+\mathbf{v}_1(\mathbf{v}_1^T n\lambda_1\mathbf{v}_1)\mathbf{v}_1^T\right] && (12.48)\\
&= \frac{1}{n}\left[\mathbf{X}^T\mathbf{X}-n\lambda_1\mathbf{v}_1\mathbf{v}_1^T-n\lambda_1\mathbf{v}_1\mathbf{v}_1^T+n\lambda_1\mathbf{v}_1\mathbf{v}_1^T\right] && (12.49)\\
&= \frac{1}{n}\mathbf{X}^T\mathbf{X}-\lambda_1\mathbf{v}_1\mathbf{v}_1^T && (12.50)
\end{aligned}$$

2. Since $\tilde{\mathbf{X}}$ lives in the $d-1$ subspace orthogonal to $\mathbf{v}_1$, the vector $\mathbf{u}$ must be orthogonal to $\mathbf{v}_1$. Hence $\mathbf{u}^T\mathbf{v}_1 = 0$ and $\mathbf{u}^T\mathbf{u} = 1$, so $\mathbf{u} = \mathbf{v}_2$.

3. We have

```
function [V, lambda] = simplePCA(C, K, f)
d = length(C);
V = zeros(d,K);
for j=1:K
  [lambda(j), V(:,j) = f(C);
  C = C - lambda(j)*V(:,j)*V(:,j)'; % deflation
end
```

12.1.8 PPCA posterior

Using the results from Section **??** with the substitutions $\mathbf{\Lambda} = \mathbf{I}$, $L^{-1} = \sigma^2\mathbf{I}$, $\mathbf{A} = \mathbf{W}$, $\mathbf{b} = \boldsymbol{\mu}$, $\boldsymbol{\mu} = \mathbf{0}$, we have

$$\text{cov}\left[\mathbf{z}|\mathbf{y}\right] = (\mathbf{I}+\mathbf{W}^T\frac{1}{\sigma^2}\mathbf{w})^{-1} = \left[\frac{1}{\sigma^2}(\sigma^2\mathbf{I}+\mathbf{W}^T\mathbf{W})\right]^{-1} = \sigma^2\mathbf{M}^{-1} \quad (12.51)$$

and

$$\mathbb{E}[\mathbf{z}|\mathbf{y}] = \text{cov}\left[\mathbf{z}|\mathbf{y}\right]\left(\mathbf{W}^T\sigma^{-2}(\mathbf{x}-\boldsymbol{\mu})\right) = \mathbf{M}^{-1}\mathbf{W}^T(\mathbf{x}-\boldsymbol{\mu}) \quad (12.52)$$

12.1.9 Latent semantic indexing

Using the code below, I find that the angles between the query and the documents are

```
   26.5830   39.9344   27.7366  140.3943   97.6569  119.5270  119.4842  113.1596   63.2745
```

and hence the top 3 documents are 1, 3, and 2.

Listing 12.1: lsiCode.m

```
% lsiCode
clear all

X = load('lsiMatrix.txt')';

%fid = fopen('lsiWords.txt');
%tmp = textscan(fid,'%s');
%fclose(fid);
%words = tmp{1};
words = textread('lsiWords.txt', '%s');


[U,S,V] = svd(X);
K = 2;
UK = U(:,1:K);
SK = S(1:K, 1:K);
VK = V(:,1:K);

[nwords ndoc] = size(X);
```

```
% plot documents in latent space
Xhat = VK';
figure(1);clf
for j=1:ndoc
  plot(Xhat(1,j), Xhat(2,j),  'o', 'linewidth', 2);
  hold on
  eps = 0.005;
  h=text(Xhat(1,j)+eps, Xhat(2,j)+eps,  sprintf('%d', j),'fontsize',18);
end

% find closest documents to query
ndx = strmatch('abducted',words);
q = zeros(nwords,1);
q(ndx) = 1;
qhat = inv(SK)*UK'*q;
for j=1:ndoc
  tmp = (qhat'*Xhat(:,j))/(norm(qhat)*norm(Xhat(:,j)));
  angle(j) = acos(tmp)*(180/pi);
end
[ndx, angles] = sort(angle)
top3 = ndx(1:3)


%Xhat = UK; % 460x2
%figure(1);clf
%for i=1:nwords
%  %plot(Xhat(i,1), Xhat(i,2),  'o');
%  h=text(Xhat(i,1), Xhat(i,2),  sprintf('%d', i),'fontsize',10);
%  hold on
%end
```

12.1.10 Imputation in a FA model

Letting $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \mathbf{\Psi}$, we have

$$p(\mathbf{x}_h|\mathbf{x}_v, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_h|\boldsymbol{\mu}_h + \mathbf{C}_{hv}\mathbf{C}_{vv}^{-1}(\mathbf{x}_v - \boldsymbol{\mu}_v), \mathbf{C}_{hh} - \mathbf{C}_{hv}\mathbf{C}_{vv}^{-1}\mathbf{C}_{vh}) \tag{12.53}$$

12.1.11 Efficiently evaluating the PPCA density

Since $\mathbf{C}$ is not full rank, we can use matrix inversion lemma to invert it efficiently:

$$\mathbf{C}^{-1} = \frac{1}{\sigma^2}\left[\mathbf{I} - \mathbf{W}\left(\mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}\right)\mathbf{W}^T\right] \tag{12.54}$$

Plugging in the MLE we find

$$\mathbf{W} = \mathbf{U}_K(\mathbf{\Lambda}_K - \sigma^2\mathbf{I})^{\frac{1}{2}} \tag{12.55}$$

$$\mathbf{C}^{-1} = \frac{1}{\sigma^2}\left[\mathbf{I} - \mathbf{U}_K(\mathbf{\Lambda}_K - \sigma^2\mathbf{I})\mathbf{\Lambda}_K^{-1}\mathbf{U}_K^T\right] \tag{12.56}$$

$$= \frac{1}{\sigma^2}\left[\mathbf{I} - \mathbf{U}_K\mathbf{J}\mathbf{U}_K^T\right] \tag{12.57}$$

$$\mathbf{J} = \mathrm{diag}(1 - \sigma^2/\lambda_j) \tag{12.58}$$

Similarly it can be shown that

$$\log|\mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}| = (d-K)\log\sigma^2 + \sum_{i=1}^{K}\log\lambda_i \tag{12.59}$$

This is implemented in the function `ppcaLoglik`.

12.1.12 PPCA vs FA

Not solved yet.

Chapter 13

Sparse linear models

13.1 Solutions

13.1.1 Partial derivative of the RSS

1. We have

$$RSS(\mathbf{w}) = \sum_i (r_i - w_k x_{ik})(r_i - w_k x_{ik}) \tag{13.1}$$

$$= \sum_i [r_i^2 - w_k^2 x_{ik}^2 - 2 w_k x_{ik} r_i] \tag{13.2}$$

So

$$\frac{\partial}{\partial w_k} RSS(\mathbf{w}) = \sum_i [-2 w_k x_{ik}^2 - 2 x_{ik} r_i] \tag{13.3}$$

Let us check this matches the standard expression

$$\nabla_{\mathbf{w}} = 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y} \tag{13.4}$$

Recall that $[\mathbf{X}^T\mathbf{X}]_{jk} = \sum_i x_{ij} x_{ik}$. For simplicity, suppose there are two features. Then

$$[\nabla_{\mathbf{w}}]_1 = 2[\sum_i x_{i1}^2, \sum_i x_{i1} x_{i2}]^T \mathbf{w} - 2[\sum_i x_{i1} y_i] = 2 w_1 \sum_i x_{i1}^2 + 2 \sum_i x_{i1}(w_2 x_{i2} - y_i) = a_1 w_1 - c_1 \tag{13.5}$$

2. Setting the derivative to zero, we have

$$w_k (\sum_i x_{ik}^2) = \sum_i (x_{ik} r_i) \tag{13.6}$$

so

$$w_k = \frac{\mathbf{x}_{:,k}^T \mathbf{r}}{\mathbf{x}_{:,k}^T \mathbf{x}_{:,k}} \tag{13.7}$$

13.1.2 Derivation of M step for EB for linear regression

We have

$$Q = \frac{1}{2}\left(N \log \beta - \beta \left(||\mathbf{y} - \mathbf{X}\mathbf{m}||^2 + \mathrm{tr}(\mathbf{X}^T\mathbf{X}\mathbf{\Sigma}) \right) + \sum_j \log \alpha_j - \mathrm{tr}[\mathbf{A}(\mathbf{m}\mathbf{m}^T + \mathbf{\Sigma})] \right) + \text{const} \tag{13.8}$$

Using $\frac{d}{d\mathbf{A}} \mathrm{tr}(\mathbf{A}\mathbf{B}) = \mathbf{B}$ we have

$$\frac{dQ}{d\alpha_j} = \frac{1}{2}\frac{1}{\alpha_j} - \frac{1}{2}(m_j^2 + \Sigma_{jj}) = 0 \tag{13.9}$$

and hence

$$\alpha_j = \frac{1}{m_j^2 + \Sigma_{jj}} \tag{13.10}$$

Also,

$$\frac{dQ}{d\beta} = \frac{N}{2}\frac{1}{\beta} - \frac{1}{2}(||\mathbf{y} - \mathbf{X}\mathbf{m}||^2 + \mathrm{tr}[\mathbf{X}^T\mathbf{X}\mathbf{\Sigma}]) = 0 \tag{13.11}$$

Hence

$$\frac{N}{\beta} = ||\mathbf{y} - \mathbf{X}\mathbf{m}||^2 + \mathrm{tr}[\beta^{-1}(\mathbf{I} - \mathbf{A}\mathbf{\Sigma})] \tag{13.12}$$

$$= ||\mathbf{y} - \mathbf{X}\mathbf{m}||^2 + \beta^{-1}\sum_j (1 - \alpha_j \Sigma_{jj}) \tag{13.13}$$

$$= ||\mathbf{y} - \mathbf{X}\mathbf{m}||^2 + \beta^{-1}\sum_j \gamma_j \tag{13.14}$$

13.1.3 Derivation of fixed point updates for EB for linear regression

Not done yet.

13.1.4 Marginal likelihood for linear regression

Derivation not done yet.

13.1.5 Reducing elastic net to lasso

We have

$$J_1(\mathbf{w}) = \mathbf{y}^T\mathbf{y} + (\mathbf{X}\mathbf{w})^T(\mathbf{X}\mathbf{w}) - 2\mathbf{y}^T(\mathbf{X}\mathbf{w}) + \lambda_2\mathbf{w}^T\mathbf{w} + \lambda_1|\mathbf{w}|_1 \tag{13.15}$$

and

$$J_2(\mathbf{w}) = \mathbf{y}^T\mathbf{y} + c^2(\mathbf{X}\mathbf{w})^T(\mathbf{X}\mathbf{w}) - 2c^2\mathbf{y}^T(\mathbf{X}\mathbf{w}) + \lambda_2 c^2\mathbf{w}^T\mathbf{w} + c\lambda_1|\mathbf{w}|_1 = J_1(c\mathbf{w}) \tag{13.16}$$

13.1.6 Shrinkage in linear regression

1. From the book, we have that $\hat{w}_k^{OLS} = \mathbf{x}_k^T\mathbf{y} = c_k/2$. Hence the solid line (1) is OLS and has slope 1/2. Also, $\hat{w}_k^{ridge} = \hat{w}_k^{OLS}/(1+\lambda_2) = \frac{c_k}{2(1+\lambda_2)}$. Hence the dotted line (2) is ridge and has slope 1/4. Finally, lasso must be the dashed line (3), since it sets coefficients to zero if $-\lambda_1 \leq c_k \leq \lambda_1$.

2. $\lambda_1 = 1$

3. $\lambda_2 = 1$

13.1.7 Prior for the Bernoulli rate parameter in the spike and slab model

Alternatively, we can put a prior on each π_j and infer the sparsity level from data. If we use a conjugate Beta prior, $\pi_j \sim \mathrm{Beta}(\alpha_1, \alpha_2)$, then we can integrate out the π_j to yield

$$p(\boldsymbol{\gamma}|\boldsymbol{\alpha}) = \int \prod_{j=1}^{D} \mathrm{Ber}(\gamma_j|\pi_j)\mathrm{Beta}(\pi_0|\alpha_1, \alpha_2)d\pi_j = \frac{B(\alpha_1 + ||\boldsymbol{\gamma}||_0, \alpha_2 + D - ||\boldsymbol{\gamma}||_0)}{B(\alpha_1, \alpha_2)} \tag{13.17}$$

The disadvantage of this is that we now have to specify α_1 and α_2 instead of just π_0. The advantage is that the results are usually less sensitive to parameters higher up in the hierarchy. Also, there is some borrowing of statistical strength between the π_j's.

13.1.8 Deriving E step for GSM prior

For a GSM prior, the posterior $p(\tau_j^2|w_j)$ turns out to be a generalized inverse Gaussian distribution. However, all we need to compute is $\mathbb{E}\left[\frac{1}{\tau_j^2}|w_j\right]$, which is given by

$$\mathbb{E}\left[\frac{1}{\tau_j^2}|w_j\right] = \int \frac{1}{\tau_j^2} p(\tau_j^2|w_j) d\tau_j^2 \tag{13.18}$$

$$= \int \frac{1}{\tau_j^2} \frac{p(\tau_j^2)p(w_j|\tau_j^2)}{p(w_j)} d\tau_j^2 \tag{13.19}$$

$$= \frac{\int \frac{1}{\tau_j^2} p(\tau_j^2)\mathcal{N}(w_j|0,\tau_j^2)d\tau_j^2}{\int p(\tau_j^2)\mathcal{N}(w_j|0,\tau_j^2)d\tau_j^2} \tag{13.20}$$

where w_j are fixed by the previous M step. Using hints 1 and 2 we have

$$\mathbb{E}\left[\frac{1}{\tau_j^2}|w_j\right] = \frac{\int -\frac{1}{|w_j|}\frac{\partial}{\partial|w_j|}\mathcal{N}(w_j|0,\tau_j)p(\tau_j^2)d\tau_j^2}{\int p(\tau_j^2)\mathcal{N}(w_j|0,\tau_j^2)d\tau_j^2} \tag{13.21}$$

$$= -\frac{1}{|w_j|}\frac{\frac{\partial}{\partial|w_j|}(\int \mathcal{N}(w_j|0,\tau_j^2)p(\tau_j^2)d\tau_j^2)}{\int p(\tau_j^2)\mathcal{N}(w_j|0,\tau_j^2)d\tau_j^2} \tag{13.22}$$

$$= -\frac{1}{|w_j|}\frac{\frac{\partial}{\partial|w_j|}p(w_j)}{p(w_j)} \tag{13.23}$$

$$= -\frac{1}{|w_j|}\frac{\partial}{\partial|w_j|}\log p(w_j) \tag{13.24}$$

$$= \frac{\pi'(w_j)}{|w_j|} \tag{13.25}$$

13.1.9 EM for sparse probit regression with Laplace prior

It is straightforward to extend the EM algorithm from Section **??** for fitting binary probit regression to handle the Laplace prior as outlined above. The main difference is that in the E step, we have to compute $\mathbb{E}\left[1/\tau_j^2\right]$ and $\mathbb{E}\left[z_i\right]$, and in the M step, we use $\mathbb{E}\left[\mathbf{z}\right]$ as the response instead of $\mathbf{y}$. See (Figueiredo 2003; Ding and Harrison 2010) for details.

13.1.10 GSM representation of group lasso

Tbhe log prior becomes

$$\log p(\mathbf{w}_g) = \frac{1}{2}\log(u_g) - \rho u_g - \frac{1}{2}\log(\rho y u_g) \tag{13.26}$$

So the corresponding NLL plus negative log prior has the form

$$J(\mathbf{w}) = NLL(\mathbf{w}) + \rho\sum_g u_g + \text{const} \tag{13.27}$$

In the case of linear regression, we have $NLL(\mathbf{w}) = \frac{1}{2\sigma^2}||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2$, so we finally get the group lasso objective:

$$J(\mathbf{w}) = NLL(\mathbf{w}) + \lambda\sum_g ||\mathbf{w}_g||_2 \tag{13.28}$$

where $\lambda \triangleq 2\sigma^2\rho$.

13.1.11 Projected gradient descent for ℓ_1 regularized least squares

We can rewrite the objective as follows:

$$Q(\boldsymbol{\theta}_+, \boldsymbol{\theta}_-) = \frac{1}{2}||\mathbf{y} - [\mathbf{X}, -\mathbf{X}][\boldsymbol{\theta}_+; \boldsymbol{\theta}_-]||_2^2 + \lambda\mathbf{1}^T\boldsymbol{\theta}_+ + \lambda\mathbf{1}^T\boldsymbol{\theta}_- \quad \text{s.t.} \quad \boldsymbol{\theta}+ \geq 0, \boldsymbol{\theta}+ \geq 0 \tag{13.29}$$

We can rewrite this as a standard QP

$$Q(\mathbf{z}) = \mathbf{c}^T\mathbf{z} + \frac{1}{2}\mathbf{z}^T\mathbf{B}\mathbf{z} \quad \text{s.t.} \quad \mathbf{z} \geq 0 \tag{13.30}$$

by defining $\mathbf{z} = [\boldsymbol{\theta}_+; \boldsymbol{\theta}-]$, $\mathbf{c} = \lambda\mathbf{1} + [-\mathbf{X}^T\mathbf{y}; \mathbf{X}^T\mathbf{y}]$ and

$$\mathbf{B} = \begin{pmatrix} \mathbf{X}^T\mathbf{X} & -\mathbf{X}^T\mathbf{X} \\ -\mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{X} \end{pmatrix} \tag{13.31}$$

The gradient is given by

$$\mathbf{g}_k = \nabla Q(\mathbf{z}_k) = \mathbf{c} + \mathbf{B}\mathbf{z}_k \tag{13.32}$$

Let us define a descent direction of the form

$$d_{jk} = \begin{cases} g_{jk} & \text{if } z_{jk} > 0 \text{ or } g_{jk} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{13.33}$$

Then the stepsize for the update is chosen using

$$\alpha_k = \underset{\alpha}{\operatorname{argmin}}\, Q(\mathbf{z}_k - \alpha\mathbf{d}_k) = \frac{\mathbf{d}_k^T\mathbf{d}_k}{\mathbf{d}_k^T\mathbf{B}\mathbf{d}_k} \tag{13.34}$$

See (Figueiredo et al. 2007) for details.

13.1.12 Subderivative of the hinge loss function

At $x = 0$ and $x = 2$ the function is differentiable and $\partial f(0) = \{-1\}$ and $\partial f(2) = \{0\}$. At $x = 1$ there is a kink, and $\partial f(1) = [-1, 0]$.

13.1.13 Lower bounds to convex functions

Let $vg \in \partial f(\mathbf{x}_0)$. Then $f(\mathbf{x}) \geq f(\mathbf{x}_0) + \mathbf{g}^T(\mathbf{x} - \mathbf{x}_0)$ is a global affine lower bound that is tight at $\mathbf{x}_0$.

Chapter 14

Kernels

14.1 Solutions

14.1.1 Fitting an SVM classifier by hand

1. The perpendicular to the decision boundary is a line through $\boldsymbol{\phi}(x_1)$ to $\boldsymbol{\phi}(x_2)$, and hence is parallel to $\boldsymbol{\phi}(x_2) - \boldsymbol{\phi}(x_1) = (1,2,2) - (1,0,0) = (0,2,2)$. Of course, any scalar multiple of this is acceptable, e.g., $(0,1,1)$.

2. There are 2 support vectors, namely the two data points. The decision boundary will be half way between them. This midpoint is $\mathbf{m} = \frac{(1,2,2)+(1,0,0)}{2} = (1,1,1)$. The distance of each of the training points to this midpoint is

$$||\boldsymbol{\phi}(x_1) - \mathbf{m}|| = ||\boldsymbol{\phi}(x_2) - \mathbf{m}|| = ||(0,1,1)|| = \sqrt{0^2 + 1^2 + 1^2} = \sqrt{2} \tag{14.1}$$

   Hence the margin is $\sqrt{2}$.

3. We have $\mathbf{w} = (0, 1/2, 1/2)$, which is parallel to $(0,2,2)$ and has $||\mathbf{w}|| = \sqrt{(1/2)^2 + (1/2)^2} = 1/\sqrt{2}$ as required.

4. Using the equations we have

$$-1([1,0,0]^T\mathbf{w} + w_0) = 1 \tag{14.2}$$
$$+1([1,2,2]^T\mathbf{w} + w_0) = 1 \tag{14.3}$$

   Hence $w_0 = -1$.

5. We have

$$w_0 + \mathbf{w}^T\boldsymbol{\phi}(x) = -1 + (0,1/2,1/2)^T(1,\sqrt{2}x,x^2) = -1 + \frac{\sqrt{2}}{2}x + \frac{1}{2}x^2 \tag{14.4}$$

14.1.2 Linear separability

No. The regularization term trades off fit to data with "simplicity", in order to suppress the effects of outliers.

Chapter 15

Gaussian processes

15.1 Solutions

15.1.1 Reproducing property

First we compute eigendecompositions of $\kappa(\mathbf{x}_1, \cdot)$ and $\kappa(\mathbf{x}_2, \cdot)$:

$$\kappa(\mathbf{x}_1, \mathbf{x}) = \sum_i \underbrace{\lambda_i \phi_i(\mathbf{x}_1)}_{f_i} \phi_i(\mathbf{x}) \tag{15.1}$$

$$\kappa(\mathbf{x}_2, \mathbf{x}) = \sum_i \underbrace{\lambda_i \phi_i(\mathbf{x}_2)}_{g_i} \phi_i(\mathbf{x}) \tag{15.2}$$

Inserting into the definition of inner product we get

$$\langle \kappa(\mathbf{x}_1, \cdot), \kappa(\mathbf{x}_2, \cdot) \rangle_{\mathcal{H}} = \sum_{i=1}^{\infty} \frac{f_i g_i}{\lambda_i} = \sum_{i=1}^{\infty} \frac{\lambda_i \phi_i(\mathbf{x}_1) \lambda_i \phi_i(\mathbf{x}_2)}{\lambda_i} = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}_1) \phi_i(\mathbf{x}_2) = \kappa(\mathbf{x}_1, \mathbf{x}_2) \tag{15.3}$$

Chapter 16

Adaptive basis function models

16.1 Solutions

16.1.1 Nonlinear regression for inverse dynamics

Below is some code by Farbood Roosta-Khorasani. He gets these numbers

```
SMSE_OLS = 0.0742

SMSE_RBFN = 0.0420 with bestParam = [k=100    sigma=25]

SMSE_NN = 0.0239 bestParam = [hidden=1000    lambda = 31.6]
```

His OLS result is consistent with the book, and the rest of the code is so simple it can't be wrong :)

Listing 16.1: Listing of `sarcosRobotArm` by Farbood Roosta-Khorasani

```
function [SMSE_OLS, SMSE_RBFN, SMSE_NN] = sarcosRobotArm

    %% load the data and setup
    clear all
    load('sarcosData.mat')

    % Standardize the inputs so they have zero mean and unit variance on the
    % training set and standardize the test set accordingly using mu and sigma
    % of training set
    [Xtrain, mu, sigma] = standardizeCols(Xtrain);
    Xtest = standardizeCols(Xtest, mu, sigma);
    assert(approxeq(mean(Xtrain),zeros(1,size(Xtrain,2))));
    assert(approxeq(sqrt(var(Xtrain)),ones(1,size(Xtrain,2))));

    % center the outputs so they have zero mean on the training set and center
    % the test test response accordingly using mu of y training set
    [ytrain mu_ytrain] = centerCols(ytrain(:,1));
    ytest = centerCols(ytest(:,1),mu_ytrain);


    % now compute the variance of the output computed on the training set
    % since it already centered, y_bar = 0;
    sigma2 = var(ytrain,1);


    %% standard linear regression
    w = linregFitL2QR(Xtrain,ytrain,0);
    ypred = Xtest*w;
    SMSE_OLS = sum((ytest-ypred).^2)/(size(ytest,1)*sigma2)

    if 1
    %% RBF network: Using K-means clustering (using cross validation to pick K)
    % Then fit an RBF network to the data, using the mu estimated by K-means.
    % Using CV to estimate the RBF bandwidth
    K = 100:50:200;
    bandwith = 5:10:25;
    paramGrid = crossProduct(K, bandwith);
    nfolds = 3;
    lossFn = @(y, yhat)sum((y-yhat).^2);
    [model, bestParam] = fitCv(paramGrid, @rbfKernelizedFit, @linregPredict, lossFn, Xtrain, ytrain, nfolds);
    ypred = linregPredict(model, Xtest);
    SMSE_RBFN = sum((ytest-ypred).^2)/(size(ytest,1)*sigma2)
    bestParam

    %% feedforward neural network.
```

```
    % mlpRegressFitNetlab and mlpRegressPredictNetlab.
    % Use CV to pick the number of hidden units and the strength of the l2 regularizer
    nHidden = 1000;
    lambda = logspace(-1,4,5);
    paramGrid = crossProduct(nHidden, lambda);
    nfolds = 3;
    lossFn = @(yhat, y)sum((yhat-y).^2);
    [model, bestParam] = fitCv(paramGrid, @NNFitNetlabWrapper, @mlpRegressPredictNetlab, lossFn, Xtrain, ytrain, nfolds)
        ;
    ypred = mlpRegressPredictNetlab(model, Xtest);
    SMSE_NN = sum((ytest-ypred).^2)/(size(ytest,1)*sigma2)
    bestParam
    end

end


function model = rbfKernelizedFit(X, y, params)
    % Fit function wrapper
    % since X and y are from CV folds, they are not centered anymore
    number_of_prototypes = params(1);
    bandwidth = params(2);
    [prototypes, ~] = kmeansFit(X, number_of_prototypes);
    K = rbfKernel(X, prototypes' , bandwidth);
    K = [ones(size(K,1),1) K];
    w  = linregFitL2QR(K, y, 0);
    model.w0 = w(1);
    model.w  = w(2:end);
    model.kernelFn = @rbfKernel;
    model.basis = prototypes';
    model.kernelParam = bandwidth;
end

function [model, output] = NNFitNetlabWrapper(X, y, params)
    H = params(1);
    lambda = params(2);
    options.DISPLAY = 'OFF';
    [model, output] = mlpRegressFitNetlab(X, y, H, lambda,options);
end
```

Chapter 17

Markov and hidden Markov Models

17.1 Solutions

17.1.1 Derivation of Q function for HMM

If we observe the state sequences, we can write the log likelihood for a single sequence as

$$\log p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|\boldsymbol{\theta}) = \log p(z_1|\boldsymbol{\pi}) + \sum_{t=2}^{T} \log p(z_t|z_{t-1}, \mathbf{A}) + \sum_{t=1}^{T} \log p(\mathbf{x}_t|z_t, \boldsymbol{\phi}) \tag{17.1}$$

$$= \sum_{k=1}^{K} \mathbb{I}(z_1 = k) \log \pi_k + \sum_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} \mathbb{I}(z_{t-1} = j, z_t = k) \log A_{jk} \tag{17.2}$$

$$+ \sum_{t=1}^{T} \sum_{k=1}^{K} \mathbb{I}(z_t = k) \log p(\mathbf{x}_t|\boldsymbol{\phi}_k) \tag{17.3}$$

Hence summing over all N sequences we have

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{k=1}^{K} N_k^1 \log \pi_k + \sum_{j=1}^{K} \sum_{k=1}^{K} N_{jk} \log A_{jk} + \sum_{i=1}^{N} \sum_{t=1}^{T_i} \sum_{k=1}^{K} \mathbb{I}(z_t = k) \log p(\mathbf{x}_{i,t}|\boldsymbol{\phi}_k) \tag{17.4}$$

where we have defined

$$N_j^1 \triangleq \sum_{i=1}^{M} \mathbb{I}(z_{i1} = j) \tag{17.5}$$

$$N_{jk} \triangleq \sum_{i=1}^{N} \sum_{t=1}^{T_i - 1} \mathbb{I}(z_{i,t} = j, z_{i,t+1} = k) \tag{17.6}$$

17.1.2 Two filter approach to smoothing in HMMs

First we find the non-stationary time-reversed transition matrix

$$A_t^r(j, i) \triangleq p(S_t = i|S_{t+1} = j) = \frac{p(S_{t+1} = j|S_t = i)p(S_t = i)}{p(S_{t+1} = j)} = \frac{A_{i,j}\Pi_t(i)}{\Pi_{t+1}(j)} \tag{17.7}$$

Then we have

$$r_t(i) = p(S_t = i|\mathbf{x}_{t+1:T}) \tag{17.8}$$

$$= \sum_j p(S_t = i, S_{t+1} = j|\mathbf{x}_{t+1:T}) \tag{17.9}$$

$$= \sum_j p(S_t = i|S_{t+1} = j)p(S_{t+1} = j|\mathbf{x}_{t+1:T}) \tag{17.10}$$

$$= \sum_j p(S_t = i|S_{t+1} = j)\frac{p(\mathbf{x}_{t+1}|S_{t+1} = j)p(S_{t+1} = j|\mathbf{x}_{t+2:T})}{p(\mathbf{x}_{t+1}|\mathbf{x}_{t+2:T})} \tag{17.11}$$

$$\propto \sum_j A_t^r(i, j)b_{t+1}(j)r_{t+1}(j) \tag{17.12}$$

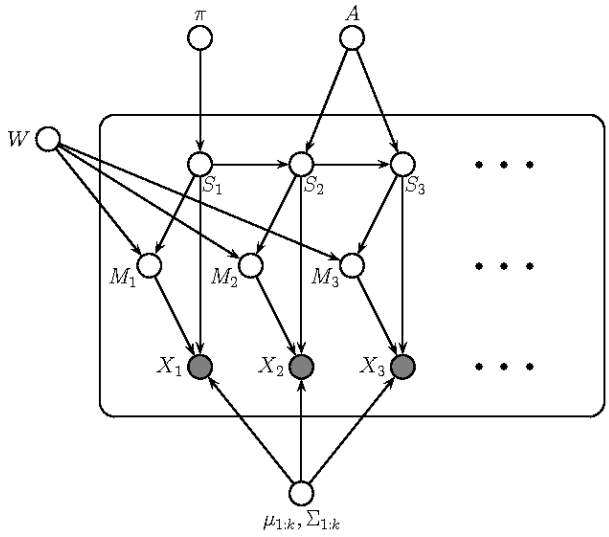


Figure 17.1: An HMM with a mixture of Gaussians observation model. M_t specifies the mixing component for time t.

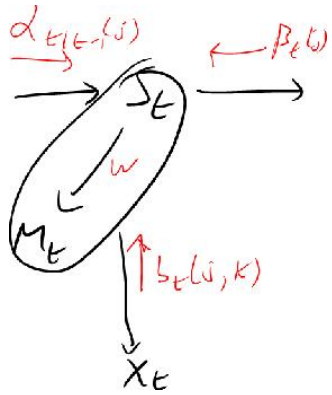


Figure 17.2: Detail on how to compute $p(S_t = j, M_t = k|\mathbf{x}_{1:T})$ after performing the forwards-backwards algorithm. We think of S_t and M_t as a single "mega-node" with internal connection strength $\mathbf{W}$. There is a bottom-up message from their shared child $\mathbf{x}_t$, and a top-down message for S_t.

In Matlab notation,

$$\mathbf{r}_t \propto (\mathbf{A}_t^r)^T(\mathbf{b}_{t+1}. * \mathbf{r}_{t+1}) \tag{17.13}$$

We can combine the two filters as follows:

$$\begin{aligned} p(S_t = i|\mathbf{x}_{1:T}) &\propto p(S_t = i|\mathbf{x}_{1:t})p(\mathbf{x}_{t+1:T}|S_t = i) && (17.14)\\ &= p(S_t = i|\mathbf{x}_{1:t})\frac{p(\mathbf{x}_{t+1:T}, S_t = i)}{p(S_t = i)} && (17.15)\\ &= p(S_t = i|\mathbf{x}_{1:t})\frac{p(S_t = i|\mathbf{x}_{t+1:T})p(\mathbf{x}_{t+1:T})}{p(S_t = i)} && (17.16)\\ &\propto \alpha_t(i)\frac{r_t(i)}{\Pi_t(i)} && (17.17) \end{aligned}$$

17.1.3 EM for for HMMs with mixture of Gaussian observations

- The model is illustrated in Figure 17.1, where m_t specifies the mixing component, so $p(m_t = k|z_t = j) = w_{jk}$.
- We can perform inference using forwards-backwards as usual, where the local evidence is given by

$$\phi_t(j) \triangleq p(\mathbf{x}_t|z_t = j) = \sum_k p(\mathbf{x}_t|z_t = j, m_t = k)p(m_t = k|z_t = j) = \sum_k w_{jk}\mathcal{N}(\mathbf{x}_t|\mu_{jk}, \mathbf{\Sigma}_{jk}) \tag{17.18}$$

To fit the model with EM, we need the joint probability

$$\gamma_t(j,k) \triangleq p(z_t = j, m_t = k|\mathbf{x}_{1:T}, \boldsymbol{\theta}) \tag{17.19}$$

One can show (Exercise **??**) that this is given by the following expression (Rabiner 1989, p267):

$$\gamma_t(j,k) = \gamma_t(j)\frac{w_{jk}\mathcal{N}(\mathbf{x}_t|\mu_{jk}, \boldsymbol{\Sigma}_{jk})}{\sum_m w_{jm}\mathcal{N}(\mathbf{x}_t|\mu_{jm}, \boldsymbol{\Sigma}_{jm})} \tag{17.20}$$

where $\gamma_t(j)$ is the usual marginal on z_t.

To compute joint probabilities of neighboring nodes in a graph, we can use the same trick that we used to compute the two-slice distribution $p(S_{t-1} = i, S_t = j|\mathbf{x}_{1:T}, \boldsymbol{\theta})$. The trick is to multiply in messages from all the neighbors, and then multiply together all the CPDs linking the nodes, as illustrated in Figure 17.2. We have

$$p(S_t = j, M_t = k|\mathbf{x}_{1:T}, \boldsymbol{\theta}) \quad \propto \quad b_t(j,k)W_{j,k}\beta_t(j)f_t(j) \tag{17.21}$$

This can be interpreted as a product of messages and potentials: f_t is a message from past evidence $\mathbf{x}_{1:t-1}$, β_t is a message from future evidence $\mathbf{x}_{t+1:T}$, $b_t(j,k)$ is a message from the current evidence $\mathbf{x}_t$, and W_{jk} is the potential which connects S_t to M_t. Normalizing this gives the following:

$$\begin{aligned} p(S_t = j, M_t = k|\mathbf{x}_{1:T}, \boldsymbol{\theta}) &= \frac{b_t(j,k)W_{j,k}\beta_t(j)f_t(j)}{\sum_{j'}\sum_{k'} b_t(j',k')W_{j',k'}\beta_t(j')f_t(j')} &(17.22)\\ &= \frac{b_t(j,k)W_{j,k}\beta_t(j)f_t(j)}{\sum_{j'} b_t(j')\beta_t(j')f_t(j')} &(17.23)\\ &= b_t(j,k)W_{j,k}\frac{\gamma_t(j)}{b_t(j)} &(17.24)\end{aligned}$$

where we used the facts that

$$\begin{aligned} \sum_{k'} b_t(j',k')W_{j',k'} &= \sum_{k'} p(\mathbf{x}_t|j',k')p(k'|j') = p(\mathbf{x}_t|j') = b_t(j') &(17.25)\\ \gamma_t(j) &= \frac{b_t(j)\beta_t(j)f_t(j)}{\sum_{j'} b_t(j')\beta_t(j')f_t(j')} &(17.26)\end{aligned}$$

We see that the final expression involves a ratio $\gamma_t(j)/b_t(j)$; the reason for this is to avoid double-counting the local evidence from $\mathbf{x}_t$.

- The M step for the transition matrix is unchanged. The M step for the observation model is as follows (assuming a single sequence for notational simplicity):

$$\begin{aligned} w_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j,k)}{\sum_{t=1}^T \sum_{m=1}^M \gamma_t(j,m)} &(17.27)\\ \boldsymbol{\mu}_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j,k)\mathbf{x}_t}{\sum_{t=1}^T \gamma_t(j,k)} &(17.28)\\ \boldsymbol{\Sigma}_{jk} &= \frac{\sum_{t=1}^T \gamma_t(j,k)(\mathbf{x}_t - \boldsymbol{\mu}_{jk})(\mathbf{x}_t - \boldsymbol{\mu}_{jk})^T}{\sum_{t=1}^T \gamma_t(j,k)} &(17.29)\end{aligned}$$

17.1.4 EM for for HMMs with tied mixtures

- For the tied-mixture HMM, we can make the observation model have the form $z_t \to m_t \to \mathbf{x}_t$. Now the hidden state emits a discrete symbol, m_t, which emits a continuous observation, $\mathbf{x}_t$. This not only reduces the number of parameters, but also reduces the number of Gaussian density calculations, which can be important to speedup inference at test time. For instance, the system can calculate $\mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for each value k of m_t, and then reuse this for each hidden state by simple reweighting. In other words, the $m_t \to \mathbf{x}_t$ arc is acting like a vector quantizer, and the $z_t \to m_t$ arc is like a dynamic reweighting of the codebook. The advantage over working with quantized data directly is that uncertainty about the codewords is maintained. See (Jelinek 1997, p161) for further details.

- We can perform inference in this model using forwards-backwards, where the local evidence is given by

$$\phi_t(j) = p(\mathbf{x}_t|z_t = j) = \sum_k p(m_t = k|z_t = j)\mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{17.30}$$

We can compute the joint distribution on (z_t, m_t) as follows:

$$\begin{aligned}
\gamma_t(j,k) &= p(z_t = j, m_t = k|\mathbf{x}_{1:T}) = p(z_t = j|\mathbf{x}_{1:T})p(m_t = k|z_t = j, \mathbf{x}_{1:T}) && (17.31)\\
&= \gamma_t(j)p(m_t = k|z_t = j, \mathbf{x}_t, \cancel{\mathbf{x}_{-t}}) && (17.32)\\
&= \gamma_t(j)\frac{p(m_t = k|z_t = j)p(\mathbf{x}_t|m_t = k)}{p(\mathbf{x}_t|z_t = j)} && (17.33)
\end{aligned}$$

From this we can compute $p(m_t = k|\mathbf{x}_{i,1:T_i}) = \sum_j \gamma_{it}(j,k)$, which replace the usual responsibilities r_{ik} when fitting a GMM.

- We can estimate $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ as in a GMM. We estimate w_{jk} as before.

Chapter 18

State space models

18.1 Solutions

18.1.1 Derivation of EM for LG-SSM

18.1.1.1 Expected complete data log likelihood

Let ℓ_c represent the expected complete data log-likelihood, which, for a single sequence, is given by

$$\ell_c(\boldsymbol{\theta}) = \log p(\mathbf{y}, \mathbf{z}|\boldsymbol{\theta}) = -\sum_{t=1}^{T}\left(\frac{1}{2}(\mathbf{y}_t - \mathbf{C}\mathbf{z}_t)^T\mathbf{R}^{-1}(\mathbf{y}_t - \mathbf{C}\mathbf{z}_t)\right) - \frac{T}{2}\log|\mathbf{R}| \tag{18.1}$$

$$-\sum_{t=2}^{T}\left(\frac{1}{2}(\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-1})^T\mathbf{Q}^{-1}(\mathbf{z}_t - \mathbf{A}\mathbf{z}_{t-1})\right) - \frac{T}{2}\log|\mathbf{Q}| \tag{18.2}$$

$$-\frac{1}{2}(\mathbf{z}_1 - \boldsymbol{\mu}_{1|0})^T\boldsymbol{\Sigma}_{1|0}^{-1}(\mathbf{z}_1 - \boldsymbol{\mu}_{1|0}) - \frac{1}{2}\log|\boldsymbol{\Sigma}_{1|0}| + \text{const} \tag{18.3}$$

It will help to rewrite this using the trace trick:

$$\ell_c(\boldsymbol{\theta}) = -\sum_{t=1}^{T}\left(\frac{1}{2}\text{tr}(\mathbf{R}^{-1}(\mathbf{y}_t\mathbf{y}_t^T + \mathbf{C}\mathbf{z}_t\mathbf{z}_t^T\mathbf{C}^T - \mathbf{C}\mathbf{z}_t\mathbf{y}_t^T - \mathbf{y}_t\mathbf{z}_t^T\mathbf{C}^T))\right) + \frac{T}{2}\log|\mathbf{R}^{-1}| \tag{18.4}$$

$$-\sum_{t=2}^{T}\left(\frac{1}{2}\text{tr}(\mathbf{Q}^{-1}(\mathbf{z}_t\mathbf{z}_t^T + \mathbf{A}\mathbf{z}_{t-1}\mathbf{z}_{t-1}^T\mathbf{A}^T - \mathbf{A}\mathbf{z}_{t-1}\mathbf{z}_t^T - \mathbf{z}_t\mathbf{z}_{t-1}\mathbf{A}^T))\right) + \frac{T}{2}\log|\mathbf{Q}^{-1}| \tag{18.5}$$

$$-\frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_{1|0}^{-1}(\mathbf{z}_1\mathbf{z}_1^T + \boldsymbol{\mu}_{1|0}\boldsymbol{\mu}_{1|0}^T - \boldsymbol{\mu}_{1|0}\mathbf{z}_1^T - \mathbf{z}_1\boldsymbol{\mu}_{1|0}^T)) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{1|0}^{-1}| + \text{const} \tag{18.6}$$

The expected complete data log likelihood is given by the following

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{i=1}^{N}\mathbb{E}_{p(\mathbf{z}|\mathbf{y}_i, \boldsymbol{\theta}^{old})}\left[\log p(\mathbf{y}_i, \mathbf{z}_i|\boldsymbol{\theta})\right] \tag{18.7}$$

where $\mathbf{y}_i = \mathbf{y}_{i,1:T_i}$ and $\mathbf{z}_i = \mathbf{z}_{i,1:T_i}$.

18.1.1.2 E step

The single-slice marginals, $\boldsymbol{\mu}_{t|T}$ and $\boldsymbol{\Sigma}_{t|T}$, can be computed using the Kalman smoother. We can compute the two-slice smoothed marginals as follows. If we define $\gamma_t = \alpha_t\beta_t$, where $\gamma_t(z) = p(\mathbf{z}_t|\mathbf{y}_{1:T})$ is the smoothed posterior, $\alpha_t(z) = p(\mathbf{z}_t|\mathbf{y}_{1:t})$ is the filtered posterior, and $\beta_t(z) \propto p(\mathbf{y}_{t+1}|\mathbf{z}_t)$ as the conditional likelihood, we can write, by analogy to Section **??**,

$$p(\mathbf{z}_t, \mathbf{z}_{t-1}|\mathbf{y}_{1:T}) \propto \alpha(\mathbf{z}_{t-1})p(\mathbf{y}_t|\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{z}_{t-1})\beta(\mathbf{z}_t) \tag{18.8}$$

One can show (Exercise 13.31 of (Bishop 2006)) that the mean of this is $\mathbb{E}\left[\mathbf{z}_t, \mathbf{z}_{t-1}|\mathbf{y}_{1:T}\right] = (\boldsymbol{\mu}_{t|T}, \boldsymbol{\mu}_{t-1|T})$, and the covariance is

$$\boldsymbol{\Sigma}_{t,t-1|T} = \text{cov}\left[\mathbf{z}_t, \mathbf{z}_{t-1}|\mathbf{y}_{1:T}\right] = \mathbf{J}_{t-1}\boldsymbol{\Sigma}_{t|T} \tag{18.9}$$

We can now compute the following expected sufficient statistics:

$$\boldsymbol{\mu}_t = \mathbb{E}\left[\mathbf{z}_t|\mathbf{y}_{1:T}\right] = \boldsymbol{\mu}_{t|T} \tag{18.10}$$

$$\mathbf{P}_t = \mathbb{E}\left[\mathbf{z}_t\mathbf{z}_t^T|\mathbf{y}_{1:T}\right] = \boldsymbol{\Sigma}_{t|T} + \boldsymbol{\mu}_{t|T}\boldsymbol{\mu}_{t|T}^T \tag{18.11}$$

$$\mathbf{P}_{t,t-1} = \mathbb{E}\left[\mathbf{z}_t\mathbf{z}_{t-1}^T|\mathbf{y}_{1:T}\right] = \boldsymbol{\Sigma}_{t,t-1|T} + \boldsymbol{\mu}_{t|T}\boldsymbol{\mu}_{t-1|T}^T \tag{18.12}$$

18.1.1.3 M step

The results below are based on (Ghahramani and Hinton 1996); we leave their detailed derivation as an exercise. In practice, it is important to regularize all these estimates to ensure numerical stability (this is even more important than in the HMM case). We leave these details as another exercise.

- Output matrix

$$\hat{\mathbf{C}} = \left(\sum_{i=1}^{N}\sum_{t=1}^{T_i}\mathbf{y}_{ti}\boldsymbol{\mu}_{ti}^T\right)\left(\sum_{i=1}^{N}\sum_{t=1}^{T_i}\mathbf{P}_{t,i}\right)^{-1} \tag{18.13}$$

- Output noise covariance

$$\hat{\mathbf{R}} = \frac{1}{T}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\left[\mathbf{y}_{ti}\mathbf{y}_{ti}^T - \hat{\mathbf{C}}\boldsymbol{\mu}_{ti}\mathbf{y}_{ti}^T - \mathbf{y}_{ti}\boldsymbol{\mu}_{ti}^T\hat{\mathbf{C}}^T + \hat{\mathbf{C}}\mathbf{P}_{ti}\hat{\mathbf{C}}^T\right] \tag{18.14}$$

$$= \frac{1}{T}\left[(\sum_{i=1}^{N}\sum_{t=1}^{T_i}\mathbf{y}_{ti}\mathbf{y}_{ti}^T) - \hat{\mathbf{C}}\left(\sum_{i=1}^{N}\sum_{t=1}^{T_i}\boldsymbol{\mu}_{ti}\mathbf{y}_{ti}^T\right)\right] \tag{18.15}$$

- State dynamics matrix

$$\hat{\mathbf{A}} = \left(\sum_{i=1}^{N}\sum_{t=2}^{T_i}\mathbf{P}_{t,t-1,i}\right)\left(\sum_{i=1}^{N}\sum_{t=2}^{T_i}\mathbf{P}_{t-1,i}\right)^{-1} \tag{18.16}$$

- State noise covariance

$$\hat{\mathbf{Q}} = \frac{1}{T-N}\sum_{i=1}^{N}\sum_{t=2}^{T_i}\left[\mathbf{P}_{t,i} - \hat{\mathbf{A}}\mathbf{P}_{t-1,t,i} - \mathbf{P}_{t,t-1,i}\hat{\mathbf{A}}^T + \hat{\mathbf{A}}\mathbf{P}_{t-1,i}\hat{\mathbf{A}}^T\right] \tag{18.17}$$

$$= \frac{1}{T-N}\left[\sum_{i=1}^{N}\sum_{t=2}^{T_i}\mathbf{P}_{t,i} - \hat{\mathbf{A}}\left(\sum_{i=1}^{N}\sum_{t=2}^{T}\mathbf{P}_{t-1,t,i}\right)\right] \tag{18.18}$$

  where $T = \sum_{i=1}^{N} T_i$ is the total length of all the sequences.

- Initial mean

$$\hat{\boldsymbol{\mu}_{1|0}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\mu}_{i,1} \tag{18.19}$$

- Initial covariance

$$\hat{\boldsymbol{\Sigma}_{1|0}} = \frac{1}{N}\sum_{i=1}^{N}\left[\mathbb{E}\left[\mathbf{z}_{1i}\mathbf{z}_{1i}^T\right] + \hat{\boldsymbol{\mu}_{1|0}}\hat{\boldsymbol{\mu}_{1|0}}^T - \hat{\boldsymbol{\mu}_{1|0}}\mathbb{E}\left[\mathbf{z}_{1i}\right]^T - \mathbb{E}\left[\mathbf{z}_{1i}\right]\hat{\boldsymbol{\mu}_{1|0}}^T\right] \tag{18.20}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left[\boldsymbol{\Sigma}_{i,1} + (\hat{\boldsymbol{\mu}_{1|0}} - \boldsymbol{\mu}_{i,1})(\hat{\boldsymbol{\mu}_{1|0}} - \boldsymbol{\mu}_{i,1})^T\right] \tag{18.21}$$

Note that the initial covariance is often set by hand to something like $\boldsymbol{\Sigma}_{1|0} = 10^{10}\mathbf{I}$, to reflect prior ignorance. In this case, we can set $\boldsymbol{\mu}_{1|0} = \mathbf{0}$.

18.1.2 Seasonal LG-SSM model in standard form

Let us initially ignore the level and trend terms. Now define $\mathbf{z}_t = (c_t^1, c_t^2, c_t^3)$; we don't need to store c_t^4, because of the sum-to-zero constraint. Let

$$\mathbf{A} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \end{pmatrix}, \ \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \ \mathbf{Q} = \begin{pmatrix} Q_c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{18.22}$$

This $\mathbf{A}$ matrix has the effect of computing the new quarter's effect from the the last three quarters' effects, and then shifting the most recent two values down the vector to reflect the update in time. For example, suppose $\boldsymbol{\mu}_{t-1} = (c^1, c^2, c^3)$. Then the update becomes

$$\boldsymbol{\mu}_t = \mathbf{A}\boldsymbol{\mu}_{t-1} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} c^1 & c^2 & c^3 \end{pmatrix} = \begin{pmatrix} -c^1 - c^2 - c^3 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} -c^4 \\ c_1 \\ c_2 \end{pmatrix} \tag{18.23}$$

Chapter 19

Undirected graphical models (Markov random fields)

19.1 Solutions

19.1.1 Derivative of the log partition function

$$\begin{aligned}
\frac{\partial \log Z(\boldsymbol{\theta})}{\partial \theta_j} &= \frac{1}{Z(\boldsymbol{\theta})} \frac{\partial Z(\boldsymbol{\theta})}{\partial \theta_j} && (19.1)\\
&= \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{x}} \frac{\partial}{\partial \theta_j} \exp[\sum_k \theta_k f_k(\mathbf{x})] && (19.2)\\
&= \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{x}} \frac{\partial}{\partial \theta_j} \prod_k \exp[\theta_k f_k(\mathbf{x})] && (19.3)\\
&= \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{x}} [\frac{\partial}{\partial \theta_j} \exp[\theta_j f_j(\mathbf{x})] \prod_{k \neq j} \exp[\theta_k f_k(\mathbf{x})] && (19.4)\\
&= \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{x}} [f_j(\mathbf{x}) \exp[\theta_j f_j(\mathbf{x})]] \prod_{k \neq j} \exp[\theta_k f_k(\mathbf{x})] && (19.5)\\
&= \frac{1}{Z(\boldsymbol{\theta})} \sum_{\mathbf{x}} f_j(\mathbf{x}) \prod_k \exp[\theta_k f_k(\mathbf{x})] && (19.6)\\
&= \sum_{\mathbf{x}} f_j(\mathbf{x}) p(\mathbf{x}) && (19.7)\\
&= \mathbb{E}\left[f_j(\mathbf{x})\right] && (19.8)
\end{aligned}$$

19.1.2 CI properties of Gaussian graphical models

1. Since there are no structural zeros in Σ, there are no marginal independencies. As for conditional independencies, we find

$$\Sigma^{-1} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

   Since $\Sigma^{-1}_{13} = 0$, we infer $X_1 \perp X_3 | X_2$, so the MRF looks like this: $X_1 - X_2 - X_3$. (If we expand out $x^T \Sigma^{-1} x$, there are no terms involving $x_1 x_3$.) This also correctly captures the fact that there are no marginal independencies, so this graph G is a perfect map for the distribution P, i.e., $I(G) = I(P)$.

2. Here $\Sigma_{13} = 0$ so we conclude $X_1 \perp X_3$. As for conditional independencies, we find

$$\Sigma^{-1} = \begin{pmatrix} 0.75 & -0.5 & 0.25 \\ -0.5 & 1.0 & -0.5 \\ 0.25 & -0.5 & 0.75 \end{pmatrix}$$

   Since there are no non-zero elements, there are no conditional independencies, so the graph is fully connected (a clique on 3 nodes). Note that the fully connected graph does not capture the marginal independence between X_1 and X_3. That is, the graph G is an I-map for the distribution ($I(G) \subset I(P)$), but not a perfect map.

3. The CPDs are

$$\begin{aligned} p(x_1) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x_2^2\right) \\ p(x_2|x_1) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_2 - x_1)^2\right) \\ p(x_3|x_2) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_3 - x_2)^2\right) \end{aligned}$$

Multiplying them together we find

$$p(x_{1:3}) = \frac{1}{(2\pi)^{3/2}} \exp\left(-\frac{1}{2}(2x_1^2 - 2x_1x_2 + 2x_2^2 - 2x_2x_3 + x_3^2)\right)$$

If we expand out the $x^T\Sigma^{-1}x$ term inside the exponent of a Gaussian we get the following (where $S = \Sigma^{-1}$):

$$x_1S_{11}x_1 + 2x_1S_{12}x_2 + 2x_1S_{13}x_3 + x_2S_{22}x_2 + 2x_2S_{23}x_3 + x_3S_{33}x_3$$

By equating $x^T\Sigma^{-1}x$ to the quadratic (in x) terms and $x^T\Sigma^{-1}\mu$ to the linear (in x) terms in $p(x_{1:3})$, we get $\mu = 0$ and

$$\Sigma^{-1} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

4. Since $\Sigma_{13}^{-1} = 0$, we see that $X_1 \perp X_3|X_2$, which is consistent with the chain-structured Bayes net. We find

$$\Sigma = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}$$

so there are no marginal independencies, which is consistent with the chain-structured Bayes net. We can always convert a directed chain to an undirected chain. $X_1 - X_2 - X_3$, without affecting the conditional indendence properties.

19.1.3 Independencies in Gaussian graphical models

1. A and B
2. C and D
3. C and D
4. A and B
5. A is true, B is false

19.1.4 Cost of training MRFs and CRFs

- For MRFs, the cost is $O(Nk + r(c+k))$. The Nk term is to compute the empirical statistics of the features. The rc term is to perform inference c times, and the rk term is to extract the expected value of each feature r times.
- For CRFs, the cost is $O(rN(c+k))$, since we must perform inference rN times, and then extract the expected features (the $O(Nk)$ time to compute the empirical expectation is negligible so is dropped from the big-O expression). In fact we can do slightly better by aggregating the marginals first, and then computing the expectations. This takes $O(Nc+k)$ time per iteration, instead of $O(Nc + Nk)$.

19.1.5 Full conditional in an Ising model

We have (dropping the conditioning on $\boldsymbol{\theta}$ for brevity)

$$p(x_i = 1|\mathbf{x}_{nb_i}) = \frac{\sum_{\mathbf{x}_{r_i}} p(x_i = 1, \mathbf{x}_{nb_i}, \mathbf{x}_{r_i})}{\sum_{x_i \in \{0,1\}} \sum_{\mathbf{x}_{r_i}} p(x_i, \mathbf{x}_{nb_i}, \mathbf{x}_{r_i})} \tag{19.9}$$

where $r_i = \{1, \ldots, n\} \setminus nb_i \setminus \{i\}$ are all the rest of the nodes, excluding i and its neighbors. Let $cl_i = nb_i \cup i$ be the closure of i. Now

$$p(x_i, \mathbf{x}_{nb_i}, \mathbf{x}_{r_i}) = \frac{1}{Z} \exp\left(x_i h_i + \sum_{j \in nb_i} J_{ij} x_i x_j + \sum_{k,j \notin cl_i} J_{kj} x_k x_j + \sum_{j \neq i} h_j x_j \right) \tag{19.10}$$

Let us define

$$S_i = h_i + \sum_{j \in nb_i} (J_{ij} x_j) \tag{19.11}$$

$$T_i = \sum_{k,j \notin cl_i} J_{kj} x_k x_j + \sum_{j \neq i} h_j x_j \tag{19.12}$$

where S_i only depends on $\mathbf{x}_{nb_i}$ and T_i only depends on $\mathbf{x}_{r_i}$. Then

$$p(x_i, \mathbf{x}_{nb_i}, \mathbf{x}_{r_i}) = \frac{1}{Z} \exp(x_i S_i) \exp(T_i) \tag{19.13}$$

When $x_i = 1$, we have $\exp(x_i S_i) = \exp(S_i)$, and when $x_i = 0$, we have $\exp(x_i S_i) = 1$. Hence

$$p(x_i = 1 | \mathbf{x}_{nb_i}) = \frac{\frac{1}{Z} \exp(S_i) \sum_{\mathbf{x}_{r_i}} \exp(T_i)}{\frac{1}{Z} \exp(S_i) \sum_{\mathbf{x}_{r_i}} \exp(T_i) + \frac{1}{Z} \sum_{\mathbf{x}_{r_i}} \exp(T_i)} \tag{19.14}$$

$$= \frac{e^{S_i}}{e^{S_i} + 1} = \frac{1}{1 + e^{-S_i}} = \sigma(S_i) \tag{19.15}$$

We see that all terms cancel except those in the Markov blanket.

If we use $x_i \in \{-1, +1\}$, we have

$$p(x_i = 1 | \mathbf{x}_{nb_i}) = \frac{\frac{1}{Z} \exp(S_i)}{\frac{1}{Z} \exp(S_i) + \frac{1}{Z} \exp(-S_i)} \tag{19.16}$$

$$= \frac{e^{S_i}}{e^{S_i} + e^{-S_i}} = \sigma(2S_i) \tag{19.17}$$

Chapter 20

Exact inference for graphical models

20.1 Solutions

20.1.1 Variable elimination

1. With $\prec = (1, 2, 3, 4, 5, 6)$, the largest intermediate term has size 3 (we connect 1,2,3 and 4,5,6).
2. See Figure 20.1(left). The largest maxclique has size 3.
3. With $\prec = (4, 1, 2, 3, 5, 6)$, the largest intermediate term has size 4 (we connect 2,3,4,5).
4. See Figure 20.1(right). The largest maxclique has size 4.

20.1.2 Gaussian times Gaussian is Gaussian

We have

$$p_1(x)p_2(x) = \frac{\sqrt{\lambda_1\lambda_2}}{2\pi}\exp\left(-(\lambda_1(x-\mu_1)^2 + \lambda_2(x-\mu_2)^2)/2\right) \quad (20.1)$$

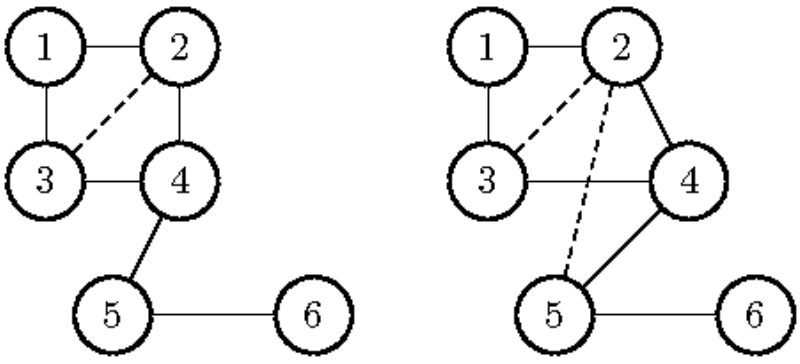
$$= \frac{C\sqrt{\lambda}}{2\pi}\exp(-\lambda(x-\mu)^2/2) \quad (20.2)$$

by completing the square.

20.1.3 Message passing on a tree

1. () The local evidence at G_2

```
v=10; mu1= 50; mu2 = 60;
x = 50; ev=[normpdf(x,mu1,sqrt(v)); normpdf(x,mu2,sqrt(v))]
```



(a) (b)

Figure 20.1: An MRF.

```
ev =
    0.1262
    0.0009
```

The potential on the $G2 - G1$ edge is $A = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$. Hence the message is

$$m_{21}(g_1) = \sum_{g_2} b_2(g_2) A(g_1, g_2) \tag{20.3}$$

which is

```
A=[0.9 0.1; 0.1 0.9]; msg50=A*ev
msg50 =
    0.1136
    0.0134
```

Technically ev above is the message from X_2 to G_2, and is not the belief at G_2. The latter is obtained by normalization, as follows:

```
bel = normalize(ev)
ans =
    0.9933
    0.0067
```

The resulting message is the same, up to a constant factor:

```
>> msg50B=A*bel
msg50B =
    0.8946
    0.1054
assert(approxeq(msg50B, normalize(msg50)))
```

The posterior on G_1 is the product of incoming messages times the local potential, which is uniform. (The messages from the unobserved childrem, G_3 and X_1, are uniform and can be ignored.)

```
>> bel = normalize([0.5;0.5] .* msg50)
bel =
ans =
    0.8946
    0.1054
```

2. () We now have two messages:

```
bel = normalize([0.5;0.5] .* msg50 .* msg50)
bel =
    0.9863
    0.0137
```

The evidence that $p(G_1 = 1)$ is stronger than before.

3. () The local evidences changes to favor state 2, which causes our belief that $G_1 = 2$ to increase. This is of course symmetric with the case $X_2 = X_3 = 50$.

```
x = 60; ev=[normpdf(x,mu1,sqrt(v)); normpdf(x,mu2,sqrt(v))]
msg60 = A*ev;
bel = normalize([0.5;0.5] .* msg60 .* msg60)
bel =
    0.0137
    0.9863
```

4. () Now G_2 is probably healthy and G_3 is probably unhealthy. By symmetry, these "forces" cancel out, and the posterior on G_3 is uniform.

```
bel = normalize([0.5;0.5] .* msg50 .* msg60)
bel =
    0.5000
    0.5000
```

Rather than computing this by hand, we can use PMTK, as follows.

Listing 20.1: :

```
G1 = 1; G2 = 2; G3 = 3;
X1 = 4; X2 = 5; X3 = 6;

graph = zeros(6);
graph(G1,X1) = 1;
graph(G1,G2) = 1;
graph(G1,G3) = 1;
graph(G2,X2) = 1;
graph(G3,X3) = 1;

% Prior on G1
CPD{G1} = TabularCPD([0.5;0.5]);

% Conditional G2|G1 and G3|G1
CPD{G2} = TabularCPD([0.9,0.1;0.1,0.9]);
CPD{G3} = CPD{G2};

% Observation model
XgivenG1 = MvnDist(50,10); % healthy
XgivenG2 = MvnDist(60,10); % unhealthy
CPD{X1}  = MvnMixDist('distributions',{XgivenG1,XgivenG2});
CPD{X2}  = CPD{X1};
CPD{X3} = CPD{X1};

dgm = DgmDist(graph,'CPDs', CPD,'infEng',VarElimInfEng(),'domain',1:6);

evidence = {[50, [50,50],  [60,60], [50,60]};
for i=1:length(evidence)
  ev = evidence{i};
  if length(ev)==2
    dgm  = condition(dgm,[X2,X3],ev);
  else
    dgm  = condition(dgm,[X2],ev);
  end
  pG1(i) = sub(pmf(marginal(dgm,G1)),1);
end
% pG1 =    0.8946  0.9863       0.0137    0.5000
```

20.1.4 Inference in 2D lattice MRFs

1. One can convert the MRF into an HMM, where the state at time t, H_t, encodes the setting of all the variables in column t, $X_{:,t}$. Hence the state space of the HMM has size 2^m. Similarly, the observation V_t encodes all the observations $Y_{:,t}$. (This is equivalent to the junction tree algorithm.)

2. The forwards-backwards algorithm takes $O(K^2T)$ time, where here $K = 2^m$ and $T = n$, so the complexity is $O(n2^{2m})$. (Variable elimination with an optimal elimination ordering would have complexity $O(n^2 2^{2m})$).

Chapter 21

Variational inference

21.1 Solutions

21.1.1 Laplace approximation to $p(\mu, \log\sigma|\mathcal{D})$ for a univariate Gaussian

Define

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2 \tag{21.1}$$

The log posterior is given by

$$\log p(\mu, \log\sigma|\mathcal{D}) = \text{const} - n\log\sigma - \frac{1}{2\sigma^2}[ns^2 + n(\overline{y}-\mu)^2] \tag{21.2}$$

For brevity, let $\lambda = \log\sigma$. The first derivatives are

$$\frac{\partial}{\partial\mu}\log p(\mu, \lambda|\mathcal{D}) = \frac{n(\overline{y}-\mu)}{\sigma^2} \tag{21.3}$$

$$\frac{\partial}{\partial\lambda}\log p(\mu, \lambda|\mathcal{D}) = -n + \frac{ns^2 + n(\overline{y}-\mu)^2}{\sigma^2} \tag{21.4}$$

from which the posterior mode is easily seen to be

$$\hat{\mu} = \overline{y}, \hat{\sigma}^2 = s^2 \tag{21.5}$$

The Hessian matrix is given by

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2}{\partial\mu^2}\log p(\mu,\lambda|\mathcal{D}) & \frac{\partial^2}{\partial\mu\partial\lambda}\log p(\mu,\lambda|\mathcal{D}) \\ \frac{\partial^2}{\partial\lambda^2}\log p(\mu,\lambda|\mathcal{D}) & \frac{\partial^2}{\partial\lambda^2}\log p(\mu,\lambda|\mathcal{D}) \end{pmatrix} = \begin{pmatrix} -\frac{n}{\sigma^2} & -2n\frac{\overline{y}-\mu}{\sigma^2} \\ -2n\frac{\overline{y}-\mu}{\sigma^2} & -\frac{2}{\sigma^2}(ns^2 + n(\overline{y}-\mu)^2) \end{pmatrix} \tag{21.6}$$

Evaluating this at the mode we have

$$\mathbf{H}|_{\hat{\boldsymbol{\theta}}} = \begin{pmatrix} -\frac{n}{\hat{\sigma}^2} & 0 \\ 0 & -2n \end{pmatrix} \tag{21.7}$$

Hence the approximate posterior is

$$p(\mu, \log\sigma|\mathcal{D}) \approx \mathcal{N}\left(\begin{pmatrix}\overline{y} \\ \log\hat{\sigma}\end{pmatrix}, \begin{pmatrix}\frac{\hat{\sigma}^2}{n} & 0 \\ 0 & \frac{1}{2n}\end{pmatrix}\right) \tag{21.8}$$

21.1.2 Laplace approximation to normal-gamma

1. For μ, we have

$$\frac{\partial}{\partial\mu}\log p(\mu, \lambda|\mathcal{D}) = \frac{2}{2\sigma^2}2n(\overline{y}-\mu) = 0 \Rightarrow \mu = \overline{y} \tag{21.9}$$

For λ, we proceed as follows. Note that $\sigma = e^{\lambda}$, so $\sigma^{-2} = e^{-2\lambda}$ and hence

$$\frac{d}{d\lambda}\sigma^{-2} = -2\sigma^{-2} \tag{21.10}$$

Thus

$$\frac{\partial}{\partial \lambda} \log p(\mu, \lambda|\mathcal{D}) = -n + \frac{1}{\sigma^2}[ns^2 + n(\overline{y} - \mu)^2] \tag{21.11}$$

2. We have

$$\frac{\partial^2}{\partial \mu^2} \log p(\mu, \lambda|\mathcal{D}) = -n/\sigma^2 \tag{21.12}$$

$$\frac{\partial^2}{\partial \mu \partial \lambda} \log p(\mu, \lambda|\mathcal{D}) = -2n\frac{\overline{y} - \mu}{\sigma^2} \tag{21.13}$$

$$\frac{\partial^2}{\partial \lambda^2} \log p(\mu, \lambda|\mathcal{D}) = \frac{-2}{\sigma^2}[ns^2 + n(\overline{y} - \mu)^2] \tag{21.14}$$

21.1.3 Variational lower bound for univariate Gaussian

For the expected log likelihood, we have

$$\mathbb{E}\left[\ln p(\mathcal{D}|\mu, \lambda)\right] = -\frac{N}{2}\log(2\pi) + \frac{N}{2}\mathbb{E}\left[\log \lambda\right]_{q(\lambda)} - \frac{\mathbb{E}\left[\lambda\right]_{q(\lambda)}}{2}\sum_{i=1}^{N}\mathbb{E}\left[(x_i - \mu)^2\right]_{q(\mu)} \tag{21.15}$$

$$= -\frac{N}{2}\log(2\pi) + \frac{N}{2}\left(\psi(a_N) - \ln b_N\right) - \frac{1}{2}\frac{a_N}{b_N}\left(\sum_i x_i^2 - 2\sum_i x_i \mu_N + N(\frac{1}{\kappa_N} + \mu_N^2)\right) \tag{21.16}$$

$$= -\frac{N}{2}\log(2\pi) + \frac{N}{2}\left(\psi(a_N) - \ln b_N\right) - \frac{1}{2}\frac{a_N}{b_N}\left(\sum_i (x_i - \mu_N)^2 + \frac{N}{\kappa_N}\right) \tag{21.17}$$

$$= -\frac{N}{2}\log(2\pi) + \frac{N}{2}\left(\psi(a_N) - \ln b_N\right) - \frac{Na_N}{2b_N}\left(\hat{\sigma}^2 + \overline{x}^2 - 2\mu_N\overline{x} + \mu_N^2 + \frac{1}{\kappa_N}\right) \tag{21.18}$$

where $\overline{x}$ and $\hat{\sigma}^2$ are the empirical mean and variance.

For the expected log prior of λ, we have

$$\mathbb{E}\left[\ln p(\lambda)\right] = (a_0 - 1)\mathbb{E}\left[\log \lambda\right] - b_0\mathbb{E}\left[\lambda\right] + a_0 \ln b_0 - \ln \Gamma(a_0) \tag{21.19}$$

$$= (a_0 - 1)(\psi(a_N) - \log b_N) - b_0\frac{a_N}{b_N} + a_0 \ln b_0 - \ln \Gamma(a_0) \tag{21.20}$$

For the expected log prior of μ, we have

$$\mathbb{E}\left[\ln p(\mu|\lambda)\right] = \frac{1}{2}\ln\frac{\kappa_0}{2\pi} + \frac{1}{2}\mathbb{E}\left[\ln \lambda\right] - \frac{1}{2}\mathbb{E}\left[(\mu - \mu_0)^2\kappa_0\lambda\right] \tag{21.21}$$

$$= \frac{1}{2}\ln\frac{\kappa_0}{2\pi} + \frac{1}{2}\left(\psi(a_N) - \ln b_N\right) - \frac{1}{2}\mathbb{E}\left[(\mu^2 - 2\mu\mu_0 + \mu_0^2)\kappa_0\lambda\right] \tag{21.22}$$

$$= \frac{1}{2}\ln\frac{\kappa_0}{2\pi} + \frac{1}{2}\left(\psi(a_N) - \ln b_N\right) - \kappa_0\frac{a_N}{b_N}\left(\frac{1}{2}(\mu_N^2 + \frac{1}{\kappa_N}) - \mu_N\mu_0 + \frac{\mu_0^2}{2}\right) \tag{21.23}$$

$$= \frac{1}{2}\ln\frac{\kappa_0}{2\pi} + \frac{1}{2}\left(\psi(a_N) - \ln b_N\right) - \frac{\kappa_0}{2}\frac{a_N}{b_N}\left[\frac{1}{\kappa_N} + (\mu_N - \mu_0)^2\right] \tag{21.24}$$

Putting the expected energy terms altogether, and using the form of the updated hyper-parameters, we get (following Robert Tseng)

$$\mathbb{E}\left[\log p(\mathcal{D}|\mu, \lambda) + \log p(\mu|\lambda) + \log p(\lambda)\right] \tag{21.25}$$

$$= \text{const} + (a_0 + \frac{N+1}{2})(\psi(a_N) + \log b_N) \tag{21.26}$$

$$-\frac{a_N}{b_N}\left[b_0 + \sum_i (x_i - \mu_N)^2 + \kappa_0(\mu_N - \mu_0)^2\right] - \frac{1}{2}\frac{1}{\kappa_N}\frac{a_N}{b_N}(\kappa_0 + N) \tag{21.27}$$

$$= \text{const} + (a_N - 1)(\psi(a_N) - \ln b_N) - \frac{a_N}{b_N}b_N - \frac{1}{2}\frac{1}{\kappa_N}\kappa_N \tag{21.28}$$

$$= \text{const} + (a_N - 1)(\psi(a_N) - \ln b_N) - a_N - a_N \ln b_N \tag{21.29}$$

$$= \text{const} - \mathbb{H}\left(q_\lambda\right) + \ln \Gamma(a_N) - a_N \ln b_N \tag{21.30}$$

Hence

$$L(q) = \text{const} + \mathbb{H}(q_\mu) + \ln \Gamma(a_N) - a_N \ln b_N \tag{21.31}$$

$$= \text{const} + \frac{1}{2} \ln \frac{1}{\kappa_N} + \ln \Gamma(a_N) - a_N \ln b_N \tag{21.32}$$

21.1.4 Derivation of the variational lower bound for VB for GMMs

See Bishop book.

21.1.5 Derivation of $\mathbb{E}[\log \pi_k]$ under a Dirichlet distribution

We write out the Dirichlet in exponential family form

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \exp\left(\sum_k \alpha_k \log \pi_k - A(\boldsymbol{\alpha})\right) \tag{21.33}$$

$$A(\boldsymbol{\alpha}) = \sum_k \log \Gamma(\alpha_k) - \log \Gamma(\sum_k \alpha_k) \tag{21.34}$$

The sufficient statistics are $\log \pi_k$,

$$\mathbb{E}[\log] \pi_k = \frac{\partial A(\boldsymbol{\alpha})}{\partial \alpha_k} = \Psi(\alpha_k) - \sum_{k'} \Psi(\alpha_{k'}) \tag{21.35}$$

where $\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x}$.

21.1.6 Alternative derivation of the mean field updates for the Ising model

An alternative way to derive the above update equations is to derive the variational free energy, and to optimize it wrt the a_i explicitly. This is the approach taken in (MacKay 2003, p425).

Let $q_i \triangleq q_i(x_i = +1)$. Using the identity that

$$\tanh(u) = 2\text{sigm}(2u) - 1 \tag{21.36}$$

we have

$$\mu_i = \tanh(a_i) = 2q_i - 1 \tag{21.37}$$

Hence we can compute q_i from a_i.

Now the entropy of q is given by

$$\mathbb{H}(q) = \sum_i H_2^{(e)}(q_i) = -q_i \ln q_i - (1 - q_i) \ln(1 - q_i) \tag{21.38}$$

where $H_2^{(e)}(q_i)$ is the binary entropy function using log base e. The average energy is

$$\langle E(\mathbf{x}) \rangle_q = \sum_{\mathbf{x}} q(\mathbf{x}) \left[-\frac{1}{2} \sum_{ij} W_{ij} x_i x_j - \sum_i L_i(x_i) \right] \tag{21.39}$$

$$= -\frac{1}{2} \sum_{i,j} W_{ij} \mu_i \mu_j - \sum_i h_i \mu_i \tag{21.40}$$

where we have assumed that $L_i(x_i) = h_i x_i$ for some constants h_i. So the variational free energy is given by

$$\tilde{F}(\mathbf{a}) = -\frac{1}{2} \sum_{i,j} W_{ij} \mu_i \mu_j - \sum_i h_i \mu_i - \sum_i H_2^{(e)}(q_i) \tag{21.41}$$

If $q_i = 1/1(1 + e^{-2a_i})$, the derivative of the entropy is

$$\frac{\partial}{\partial q_i} H_2^{(e)}(q_i) = \ln \frac{1 - q_i}{q_i} = -2a_i \tag{21.42}$$

Hence

$$\frac{\partial}{\partial a_i}\tilde{F}(\mathbf{x}) = \left[-\sum_j W_{ij}\mu_j - h_i\right]\left(2\frac{\partial q_i}{\partial a_i}\right) - \ln\left(\frac{1-q_i}{q_i}\right)\left(\frac{\partial q_i}{\partial a_i}\right) \quad (21.43)$$

$$= \left(2\frac{\partial q_i}{\partial a_i}\right)\left[-\left(\sum_j W_{ij}\mu_i + h_i\right) + a_i\right] \quad (21.44)$$

Setting the derivative to zero gives the update

$$a_i = \sum_j W_{ij}\mu_j + h_j \quad (21.45)$$

where $\mu_j = \tanh(a_j)$. This can be rearranged to give Equation **??**.

21.1.7 Forwards vs reverse KL divergence

(We follow the presentation of Paul Vanetti.)

1. We have

$$\mathbb{KL}\left(p||q\right) = \sum_{xy} p(x,y)[\log p(x,y) - \log q(x) - \log q(y)] \quad (21.46)$$

$$= \sum_{xy} p(x,y)\log p(x,y) - \sum_x p(x)\log q(x) - \sum_y p(y)\log q(y) \quad (21.47)$$

   We can optimize wrt $q(x)$ and $q(y)$ separately. Imposing a Lagrange multiplier to enforce the constraint that $\sum_x q(x) = 1$ we have the Lagrangian

$$\mathcal{L}(q,\lambda) = \sum_x p(x)\log q(x) + \lambda(1 - \sum_x q(x)) \quad (21.48)$$

   Taking derivatives wrt $q(x)$ (thinking of the function as a finite length vector, for simplicity), we have

$$\frac{\partial \mathcal{L}}{\partial q(x)} = \frac{p(x)}{q(x)} - \lambda = 0 \quad (21.49)$$

$$q(x) = \frac{p(x)}{\lambda} \quad (21.50)$$

   Summing both sides over x we get $\lambda = 1$ and hence

$$q(x) = p(x) \quad (21.51)$$

   Analogously, $q(y) = p(y)$.

2. We require $q(x,y) = 0$ whenever $p(x,y) = 0$, otherwise $\log q(x,y)/p(x,y) = \infty$. Since $q(x,y) = q_x(x)q_y(y)$, it must be that $q_x(x) = q_y(y)$ whenever $x = y$, and hence $q_x = q_y$ are the same distribution. There are only 3 possible distributions that put 0s in the right places and yet sum to 1. The first is:

		x				
		1	2	3	4	q(y)
	1	0	0	0	0	0
y	2	0	0	0	0	0
	3	0	0	1	0	1
	4	0	0	0	0	0
	q(x)	0	0	1	0	

   The second one is

		x				
		1	2	3	4	q(y)
	1	0	0	0	0	0
y	2	0	0	0	0	0
	3	0	0	0	0	0
	4	0	0	0	1	1
	q(x)	0	0	0	1	

For both of these, we have $\mathbb{KL}\left(q||p\right) = 1 \times \log \frac{1}{1/4} = \log 4$. Furthermore, any slight perturbation of these probabilities away from the designated values will cause the KL to blow up, meaning these are local minima.

The final local optimum is

		x				
		1	2	3	4	q(y)
y	1	1/4	1/4	0	0	1/2
	2	1/4	1/4	0	0	1/2
	3	0	0	0	0	0
	4	0	0	0	0	0
	q(x)	1/2	1/2	0	0	

This has $\mathbb{KL}\left(q||p\right) = 4(\frac{1}{4}\log\frac{1/4}{1/8}) = \log 2$, so this is actually the global optimum.

To see that there are no other solutions, one can do a case analysis, and see that any other distribution will not put 0s in the right places. For example, consider this:

		x				
		1	2	3	4	q(y)
y	1	1/4	0	1/4	0	1/2
	2	0	0	0	0	0
	3	1/4	0	1/4	0	1/2
	4	0	0	0	0	0
	q(x)	1/2	0	1/2	0	

Obviously if we set $q(x,y) = p(x)p(y) = 1/16$, we get $\mathbb{KL}\left(q||p\right) = \infty$.

21.1.8 Derivation of the structured mean field updates for FHMM

Let $\overline{\mathbf{x}}_{tm} = \mathbb{E}_q\left[\mathbf{x}_{tm}\right]$. We can write the KL divergence as

$$\begin{aligned}
\text{KL} &= \sum_{t=1}^{T} \overline{\mathbf{x}}_{tm}^T \tilde{\boldsymbol{\xi}}_{tm} + \frac{1}{2}\sum_{t=1}^{T}\left[\mathbf{y}_t^T\boldsymbol{\Sigma}^{-1}\mathbf{y}_t - 2\sum_{m=1}^{M}\mathbf{y}_t\boldsymbol{\Sigma}^{-1}\mathbf{W}_m\overline{\mathbf{x}}_{tm}\right. && (21.52)\\
&\quad \left. + \sum_{m=1}^{M}\sum_{n\neq m}^{M}\text{tr}\left(\mathbf{W}_m^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_n\overline{\mathbf{x}}_{tn}\overline{\mathbf{x}}_{tm}^T\right) + \sum_{m=1}^{M}\text{tr}\left(\mathbf{W}_m^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_m\text{diag}(\overline{\mathbf{x}}_{tm})\right)\right] && (21.53)\\
&\quad - \log Z_q + \log Z && (21.54)
\end{aligned}$$

Now since

$$\frac{\partial Z_q}{\partial\tilde{\boldsymbol{\xi}}_{tn}} = \overline{\mathbf{x}}_{tn} \tag{21.55}$$

we have

$$\frac{\partial\text{KL}}{\partial\tilde{\boldsymbol{\xi}}_{tn}} = \overline{\mathbf{x}}_{tn} + \sum_{t=1}^{T}\sum_{m=1}^{M}\left[\tilde{\boldsymbol{\xi}}_{tm} - \mathbf{W}_m^T\boldsymbol{\Sigma}^{-1}\mathbf{y}_t + \sum_{\ell\neq m}^{M}\mathbf{W}_m^T\boldsymbol{\Sigma}^{-1}\mathbf{W}_\ell\overline{\mathbf{x}}_{t,\ell} + \frac{1}{2}\boldsymbol{\delta}_m\right]\frac{\partial\overline{\mathbf{x}}_{tm}}{\partial\tilde{\boldsymbol{\xi}}_{tn}} - \overline{\mathbf{x}}_{tn} \tag{21.56}$$

Setting the term inside the brackets to zero gives

$$\boldsymbol{\xi}_{tm} = \exp\left(\mathbf{W}_m^T\mathbf{C}^{-1}\tilde{\mathbf{y}}_{tm} - \frac{1}{2}\boldsymbol{\delta}_m\right) \tag{21.57}$$

where

$$\tilde{\mathbf{y}}_{tm} \triangleq \mathbf{y}_t - \sum_{\ell\neq m}^{M}\mathbf{W}_\ell\overline{\mathbf{x}}_{t,\ell} \tag{21.58}$$

21.1.9 Variational EM for binary FA with sigmoid link

The approach we will take, following (Tipping 1998), is to use the quadratic lower bound to the sigmoid function described in Section **??**. This will convert the logistic likelihood into a Gaussian likelihood, which will allow us to fit the model as if it were a Gaussian FA model.

Based on Section **??**, we have the following lower bound on the likelihood:

$$p(\mathbf{x}_i|\mathbf{z}_i,\boldsymbol{\theta}) \geq s(\boldsymbol{\xi}_i)\mathcal{N}(\tilde{\mathbf{x}}_i|\tilde{\mathbf{W}}\tilde{\mathbf{z}}_i,\mathbf{A}_i^{-1}) = s(\boldsymbol{\xi}_i)\mathcal{N}(\tilde{\mathbf{x}}_i|\mathbf{W}\mathbf{z}_i+\boldsymbol{\beta},\mathbf{A}_i^{-1}) \tag{21.59}$$

$$\tilde{\mathbf{x}}_i \triangleq \mathbf{A}_i^{-1}(\mathbf{b}_i+\mathbf{x}_i) \tag{21.60}$$

$$\tag{21.61}$$

where $s(\boldsymbol{\xi}_i)$ is a constant factor independent of $\mathbf{z}_i$ and $\boldsymbol{\theta}$. (Recall that $\mathbf{A}_i$, $\mathbf{b}_i$, and hence are all functions of $\boldsymbol{\xi}_i$, although we have suppressed this in the notation.) We can combine this Gaussian likelihood with a Gaussian prior $p(\mathbf{z}_i) = \mathcal{N}(\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)$ to get a Gaussian posterior of the form $p(\mathbf{z}_i|\mathbf{x}_i,\boldsymbol{\theta}) \approx \mathcal{N}(\mathbf{z}_i|\mathbf{m}_i,\boldsymbol{\Sigma}_i)$, where

$$\boldsymbol{\Sigma}_i^{-1} = \mathbf{W}^T\mathbf{A}_i\mathbf{W} + \boldsymbol{\Sigma}_0^{-1} = \boldsymbol{\Sigma}_0^{-1} + 2\sum_j \lambda_{ij}\mathbf{w}_j\mathbf{w}_j^T \tag{21.62}$$

$$\mathbf{m}_i = \boldsymbol{\Sigma}_i\left(\mathbf{W}^T\mathbf{A}_i(\tilde{\mathbf{x}}_i-\boldsymbol{\beta}) + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right) = \boldsymbol{\Sigma}_i\left(\sum_j\left[x_{ij}-\frac{1}{2}-2\lambda_{ij}\beta_j\right]\mathbf{w}_j + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right) \tag{21.63}$$

This result is identical to the Gaussian FA case, except we replace $\mathbf{x}_i$ with $\tilde{\mathbf{x}}_i$ and we replace $\boldsymbol{\Psi}$ with $\mathbf{A}_i^{-1}$. The normalization constant for the approximate posterior is given by Equation **??**, which in this case yields

$$p(\mathbf{x}_i|\boldsymbol{\theta}) \approx \mathcal{N}(\tilde{\mathbf{x}}_i|\mathbf{W}\boldsymbol{\mu}_0+\boldsymbol{\beta},\mathbf{A}_i^{-1}+\mathbf{W}\boldsymbol{\Sigma}_0\mathbf{W}^T) \tag{21.64}$$

Of course, the likelihood and hence the posterior depends on the variational parameters $\mathbf{x}_i$. We start with an initial guess and we then update the variational parameters to maximize the lower bound, as in Equation **??**. The result is

$$\xi_{ij}^2 = \tilde{\mathbf{w}}_j^T\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right]\tilde{\mathbf{w}}_j = \begin{pmatrix}\mathbf{w}_j^T & \beta_j\end{pmatrix}\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right]\begin{pmatrix}\mathbf{w}_j\\ \beta_j\end{pmatrix} = \mathbf{w}_j^T(\boldsymbol{\Sigma}_i+\mathbf{m}_i\mathbf{m}_i^T)\mathbf{w}_j + 2\beta_j\mathbf{w}_j^T\mathbf{m}_i + \beta_j^2 \tag{21.65}$$

where we used the fact that

$$\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right] = \begin{pmatrix}(\boldsymbol{\Sigma}_i+\mathbf{m}_i\mathbf{m}_i^T) & \mathbf{m}_i\\ \mathbf{m}_i^T & 1\end{pmatrix} \tag{21.66}$$

We can initialize using the prior parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ instead of the posterior parameters $\mathbf{m}_i$ and $\boldsymbol{\Sigma}_i$. This procedure usually converges in a small number of iterations, this completing the E step.

For the M step, we cannot just re-use Equation **??** for Gaussian FA, since the measurement noise is not constant. However, using the trace trick we have

$$\mathbb{E}\left[(\tilde{\mathbf{y}}_i-\tilde{\mathbf{W}}\tilde{\mathbf{z}}_i)^T\mathbf{A}_i(\tilde{\mathbf{y}}_i-\tilde{\mathbf{W}}\tilde{\mathbf{z}}_i)\right] = \mathrm{tr}(\mathbf{A}_i\mathbf{G}_i(\tilde{\mathbf{W}})) \tag{21.67}$$

$$\mathbf{G}_i(\tilde{\mathbf{W}}) \triangleq \tilde{\mathbf{y}}_i\tilde{\mathbf{y}}_i^T + \tilde{\mathbf{W}}\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right]\tilde{\mathbf{W}}^T - 2\tilde{\mathbf{W}}\mathbb{E}\left[\tilde{\mathbf{z}}_i\right]\tilde{\mathbf{y}}_i^T \tag{21.68}$$

Hence

$$Q = -\frac{1}{2}\sum_i \mathrm{tr}(\mathbf{A}_i\mathbf{G}(\tilde{\mathbf{W}})) + \mathrm{const} \tag{21.69}$$

$$\nabla_{\tilde{\mathbf{W}}}Q = -\frac{1}{2}\sum_i \mathbf{A}_i\left[2\tilde{\mathbf{W}}\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right] - 2\tilde{\mathbf{y}}_i\mathbb{E}\left[\tilde{\mathbf{x}}_i^T\right]\right] = 0 \tag{21.70}$$

$$\tilde{\mathbf{W}} = (\sum_i \mathbf{A}_i\tilde{\mathbf{y}}_i\mathbb{E}\left[\tilde{\mathbf{z}}_i^T\right])(\sum_i \mathbf{A}_i\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right])^{-1} \tag{21.71}$$

$$\tilde{\mathbf{w}}_j = (\sum_i a_{ij}\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right])^{-1}(\sum_i a_{ij}\tilde{y}_{ij}\mathbb{E}\left[\tilde{\mathbf{z}}_i\right]) \tag{21.72}$$

The final result follows from the fact that $a_{ij} = 2\lambda_{ij}$ and $\tilde{y}_{ij} = \frac{y_{ij}-\frac{1}{2}}{a_{ij}}$.

So the M step has the following form:

$$\tilde{\mathbf{w}}_j = \left[\sum_i 2\lambda_{ij}\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right]\right]^{-1}\left[\sum_i(x_{ij}-\frac{1}{2})\mathbb{E}\left[\tilde{\mathbf{z}}_i\right]^T\right] \tag{21.73}$$

where $\mathbb{E}\left[\tilde{\mathbf{z}}_i\right] = [\mathbf{m}_i;1]$ and $\mathbb{E}\left[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^T\right]$ is given in Equation 21.66. We can estimate $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ using Equations **??** and **??** respectively; there is no need to change anything since the prior is the same. Of course, we no longer need to estimate $\boldsymbol{\Psi}$.

21.1.10 VB for binary FA with probit link

See (Rogers and Girolami 2011, p266).

Chapter 22

More variational inference

Chapter 23

Monte Carlo inference

23.1 Solutions

23.1.1 Sampling from a Cauchy

The pdf is

$$f(x) = \frac{1}{\pi}\frac{1}{1+x^2} \tag{23.1}$$

Using the fact that

$$\int \frac{1}{a^2+u^2}du = \frac{1}{\tan}^{-1}(\frac{u}{a}) + C \tag{23.2}$$

we can write the cdf as follows:

$$F(x) = \frac{1}{\pi}\tan^{-1}(x) + \frac{1}{2} \tag{23.3}$$

where we chose the constant $C = \frac{1}{2}$ to ensure the range of the cdf is $[0, 1]$.

23.1.2 Rejection sampling from a Gamma using a Cauchy proposal

Not solved yet, but most of the details (except for the value of M) are in (Bishop 2006, p530). Figure should look like his Figure 11.5.

23.1.3 Optimal proposal for particle filtering with linear-Gaussian measurement model

We have

$$p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{y}_t) = \mathcal{N}(\mathbf{m}_t, \mathbf{\Sigma}_t) \tag{23.4}$$

where

$$\mathbf{\Sigma}_t^{-1} = \mathbf{Q}_{t-1}^{-1} + \mathbf{H}_t^T\mathbf{R}_t^{-1}\mathbf{H}_t \tag{23.5}$$

$$\mathbf{m}_t = \mathbf{\Sigma}_t(\mathbf{Q}_{t-1}^{-1}\mathbf{f}_t(\mathbf{z}_{t-1}) + \mathbf{H}_t^T\mathbf{R}_t^{-1}\mathbf{y}_t) \tag{23.6}$$

Also,

$$p(\mathbf{y}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{H}_t\mathbf{f}_t(\mathbf{z}_{t-1}), \mathbf{Q}_{t-1} + \mathbf{H}_t\mathbf{R}_t\mathbf{H}_t^T) \tag{23.7}$$

Chapter 24

Markov Chain Monte Carlo (MCMC) inference

24.1 Solutions

24.1.1 Gibbs sampling from a 2D Gaussian

Recall from Section **??** that, if we partition a Gaussian random vector $\mathbf{x} = (x_1, x_2)$, then we can compute $p(x_1|x_2)$ as follows:

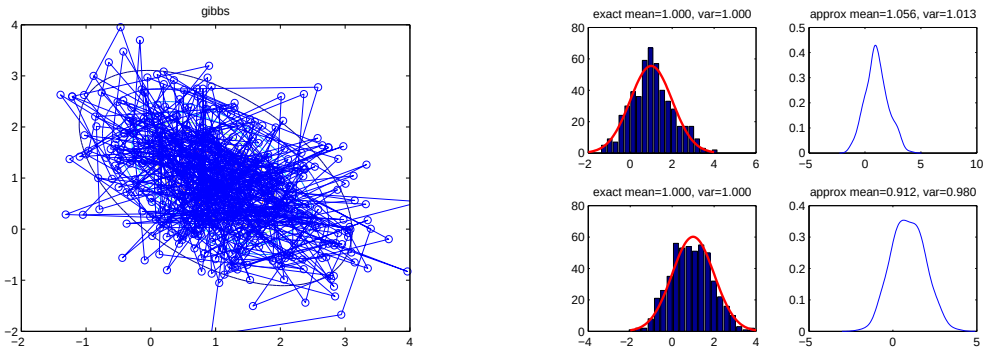
$$
\begin{aligned}
p(x_1|x_2) &= \mathcal{N}(x_1|\mu_{1|2}, \Sigma_{1|2}) && (24.1)\\
\mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) && (24.2)\\
\Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} && (24.3)
\end{aligned}
$$

The expression for $p(x_2|x_1)$ is analogous.

We can alternate between sampling x_1 (a horizontal move) and sampling x_2 (a vertical move). 500 steps of this process produce the result shown in Figure 24.1(a). We can compute marginal distributions $p(x_1)$ and $p(x_2)$ by simply "throwing away" the unwanted components of the joint vector, i.e.,

$$
p(x_i) \approx \frac{1}{S}\sum_{s=1}^{S} \delta_{x_i^s}(x_i) \tag{24.4}
$$

See `mcmcMvn2d` for the code.



(a) (b)

Figure 24.1: (a) 500 samples from a 2D Gaussian $\mathcal{N}([1,1],[1,-0.5;-0.5,1])$ generated using Gibbs sampling, with a burn-in of 500. (b) Empirical marginals of $p(x_1)$ and $p(x_2)$. Figure generated by `mcmcMvn2d`.

24.1.2 Gibbs sampling for 1D mixture of Gaussians

1. Markov blanket of π is z, α.

$$p(\pi|z,\alpha) \propto \prod_{n=1}^{N}\prod_{i=1}^{k}(\pi_i)^{z_n^i} \cdot \text{Dir}(\pi|\alpha) \tag{24.5}$$

$$\propto \prod_{i=1}^{k}(\pi_i)^{\sum_{n=1}^{N} z_n^i} \cdot \prod_{i=1}^{K}(\pi_i)^{\alpha_i - 1} \tag{24.6}$$

$$= \text{Dir}\left(\alpha_1 + \sum_{n=1}^{N} z_n^1, \ldots, \alpha_K + \sum_{n=1}^{N} z_n^K\right) \tag{24.7}$$

2. MB of μ_j is x, z, σ, η, τ.

$$p(\mu_j|x,z,\sigma,\eta,\tau) \propto \left(\prod_{n=1}^{N}\mathcal{N}(x_n|\mu_j,\sigma_j^2)^{z_n^j}\right)\mathcal{N}(\mu_j|\eta,\tau) \tag{24.8}$$

$$\propto \exp -\frac{1}{2}\left(\frac{1}{\sigma_j^2}\sum_{n=1}^{N}(z_n^j(x_n^2+\mu_j^2-2x_n\mu_j)) + \frac{1}{\tau^2}(\mu_j^2+\eta^2-2\mu_j\eta)\right) \tag{24.9}$$

$$= \exp -\frac{1}{2}\left(\mu_j^2(\frac{\sum_n z_n^j}{\sigma_j^2}+\frac{1}{\tau^2}) - 2\mu_j(\frac{\sum_n z_n^j x_n}{\sigma_j^2}+\frac{\eta}{\tau^2}) + \text{const}\right) \tag{24.10}$$

$$\triangleq \mathcal{N}(\mu_j|\hat{\mu}_j,\hat{\sigma}_j^2) \propto \exp -\frac{1}{2\hat{\sigma}_j^2}\left(\mu_j^2+\hat{\mu}_j^2-2\mu_j\hat{\mu}_j\right) \tag{24.11}$$

Matching terms in μ_j^2 we get

$$\frac{1}{\hat{\sigma}^2} = \frac{\sum_n z_n^j}{\sigma_j^2}+\frac{1}{\tau^2} \tag{24.12}$$

$$\hat{\sigma}_j^2 = \frac{\tau^2\sum_n z_n^j + \sigma_j^2}{\tau^2\sigma_j^2} \tag{24.13}$$

Matching terms for $-2\mu_j$ we get

$$\frac{\hat{\mu_j}}{\hat{\sigma_j}^2} = \frac{\sum_n z_n^j x_n}{\sigma_j^2}+\frac{\eta}{\tau^2} \tag{24.14}$$

$$\hat{\mu_j}^2 = \hat{\sigma_j}^2\left(\frac{\tau^2\sum_n z_n^j x_n + \eta\sigma_j^2}{\sigma_j^2\tau_2}\right) \tag{24.15}$$

Hence

$$p(\mu_j|x,z) = \mathcal{N}\left(\frac{\sum_n z_n^j x_n\tau^2+\eta\sigma_j^2}{\tau^2\sum_n z_n^2+\sigma_j^2}, \frac{\tau^2\sum_n z_n^j+\sigma_j^2}{\tau^2\sigma_j^2}\right) \tag{24.16}$$

3. MB for σ_j is x, z, a, b.

$$p(\sigma_j|x,z,a,b) \propto \left(\prod_{n=1}^{N}N(x_n|\mu_j,\sigma_j^2)^{z_n^j}\right)IG(\sigma_j|a,b) \tag{24.17}$$

$$\propto \sigma_j^{-\sum_{n=1}^{N} z_n^j}\exp\left(-\frac{1}{2\sigma_j^2}\sum_{n=1}^{N} z_n^j(x_n-\mu_j)^2\right)\sigma_j^{-2(a+1)}\exp\left(-\frac{b}{\sigma^2}\right) \tag{24.18}$$

$$= IG(\sigma_j^2|a_N,b_N) \tag{24.19}$$

$$a_N = a+\frac{1}{2}\sum_{n=1}^{N} z_n^j \tag{24.20}$$

$$b_N = b+\frac{\sum_{n=1}^{N} z_n^j(x_n-\mu_j)^2}{2} \tag{24.21}$$

4. MB for z_n is x_n, μ, σ, π.

$$P(z_n=j|x_n,\mu,\sigma,\pi) \propto \pi_j N(x_n|\mu_j,\sigma_j^2) \tag{24.22}$$

24.1.3 Gibbs sampling from a Potts model

Note done.

24.1.4 Full conditionals for hierarchical model of Gaussian means

Note done yet.

24.1.5 Gibbs sampling for robust linear regression with a Student t likelihood

The key observation is that the conditional likelihood is given by

$$p(\mathcal{D}|\mathbf{w}, \sigma^2, \mathbf{z}) = \mathcal{N}(\mathbf{y}|\mathbf{X}\mathbf{w}, \sigma^2\mathbf{\Sigma}) \tag{24.23}$$

where $\mathbf{\Sigma} = \mathrm{diag}(\mathbf{z})$. Hence if we use a conjugate prior, the conditional posterior is also conjugate:

$$\begin{aligned} p(\mathbf{w}, \sigma^2|\mathbf{z}, \mathcal{D}) &= \mathrm{NIG}(\mathbf{w}_n, \mathbf{V}_n, a_n, b_n) && (24.24)\\ \mathbf{w}_n &= \mathbf{V}_n(\mathbf{V}_0^{-1}\mathbf{w}_0 + \mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{y}) && (24.25)\\ \mathbf{V}_n &= (\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{\Sigma}^{-1}\mathbf{X})^{-1} && (24.26)\\ a_n &= a_0 + n/2 && (24.27)\\ b_n &= b_0 + \frac{1}{2}\left((\mathbf{w}_0^T\mathbf{V}_0^{-1}\mathbf{w}_0 + \mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y} - \mathbf{w}_n^T\mathbf{V}_n^{-1}\mathbf{w}_n\right) && (24.28) \end{aligned}$$

So we can alternate between sampling from $p(\sigma^2|\mathbf{w}, \mathbf{z}, \mathcal{D}) = \mathrm{IG}(a_n, b_n)$. $p(\mathbf{w}|\sigma^2, \mathbf{z}, \mathcal{D}) = \mathcal{N}(\mathbf{w}_n, \sigma^2\mathbf{V}_n)$, and $p(\mathbf{z}|\mathbf{y}, \mathbf{w}, \sigma^2) = \prod_i \mathrm{Ga}(\frac{\nu+1}{2}, \frac{\nu+\delta_i}{2})$. (If we wish to sample the dof parameter ν as well, we would have to add a Metropolis-Hastings step, since this parameter does not have a closed form full conditional.)

24.1.6 Gibbs sampling for probit regression

Here are the full conditionals:

- Latent variables:

$$p(z_i|y_i, \mathbf{x}_i, \mathbf{w}) = \begin{cases} \mathcal{N}(z_i|\mathbf{w}^T\mathbf{x}_i, 1)\mathbb{I}(z_i > 0) & \text{if } y_i = 1 \\ \mathcal{N}(z_i|\mathbf{w}^T\mathbf{x}_i, 1)\mathbb{I}(z_i < 0) & \text{if } y_i = 0 \end{cases} \tag{24.29}$$

- Parameters:

$$\begin{aligned} p(\mathbf{w}|\mathcal{D}, \mathbf{z}, \boldsymbol{\lambda}) &= \mathcal{N}(\mathbf{w}_N, \mathbf{V}_N) && (24.30)\\ \mathbf{V}_N &= (\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{X})^{-1} && (24.31)\\ \mathbf{w}_N &= \mathbf{V}_N(\mathbf{V}_0^{-1}\mathbf{m}_0 + \mathbf{X}^T\mathbf{z}) && (24.32) \end{aligned}$$

We can also sample the hyper-parameters, $\mathrm{diag}(\mathbf{V}_0)$, as an appealing alternative to cross validation. This is something that is not so simple to do using EM, which is restricted to one level of unknowns in the hierarchy.

24.1.7 Gibbs sampling for logistic regression with the Student approximation

- For the latent variables, the posterior is given by $p(z_i|y_i, \mathbf{x}_i, \mathbf{w}) \propto p(y_i|z_i)p(z_i|\mathbf{x}_i, \mathbf{w})$, so we see that the posterior is a truncated Gaussian:

$$p(z_i|y_i, \mathbf{x}_i, \mathbf{w}) = \begin{cases} 2\mathcal{N}(z_i|\mathbf{w}^T\mathbf{x}_i, \lambda_i^{-1})\mathbb{I}(z_i > 0) & \text{if } y_i = 1 \\ 2\mathcal{N}(z_i|\mathbf{w}^T\mathbf{x}_i, \lambda_i^{-1})\mathbb{I}(z_i < 0) & \text{if } y_i = 0 \end{cases} \tag{24.33}$$

- For the scale variables, the posterior is given by

$$p(\lambda_i|z_i, \mathbf{w}, \mathbf{x}_i, \nu) = \mathrm{Ga}\left(\frac{\nu+1}{2}, \frac{\nu + (z_i - \mathbf{w}^T\mathbf{x}_i)^2}{2}\right) \tag{24.34}$$

- For the parameter vector, assuming a prior of the form $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{V}_0)$, we have

$$\begin{aligned} p(\mathbf{w}|\mathcal{D}, \mathbf{z}, \boldsymbol{\lambda}) &= \mathcal{N}(\mathbf{w}_N, \mathbf{V}_N) && (24.35)\\ \mathbf{V}_N &= (\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{\Lambda}\mathbf{X})^{-1} && (24.36)\\ \mathbf{m}_N &= \mathbf{V}_N(\mathbf{V}_0^{-1}\mathbf{m}_0 + \mathbf{X}^T\mathbf{\Lambda}\mathbf{z}) && (24.37)\\ \mathbf{\Lambda} &= \mathrm{diag}(\lambda_i) && (24.38) \end{aligned}$$

Chapter 25

Clustering

Chapter 26

Graphical model structure learning

26.1 Solutions

26.1.1 Causal reasoning in the sprinkler network

1. We cut the arc from $S \to W$ and do inference in the resulting graph. Thus the fact that W is on does not provide evidence about the state of S, so we just use its prior.

$$p(S = T|\text{do}(W = T)) = \sum_c p(S = T|C = c)p(C = c) = 0.5 \times 0.1 + 0.5 \times 0.5 = 0.3 \tag{26.1}$$

2. Since $W \perp S$ in the mutated graph,

$$p(S = T|\text{do}(W = F)) = 0.3 \tag{26.2}$$

3. If we set $C = T$, the downstream nodes can use this as evidence

$$p(S = T|\text{do}(C = T)) = p(S = T|C = T) = 0.1 \tag{26.3}$$

Chapter 27

Latent variable models for discrete data

27.1 Solutions

27.1.1 Partition function for an RBM

$$
\begin{aligned}
Z &= \sum_{\mathbf{h}} \sum_{\mathbf{v}} \exp(\sum_r \sum_k v_r h_k W_{rk}) = \sum_{\mathbf{h}} \sum_{\mathbf{v}} \prod_r \exp(\sum_k v_r h_k W_{rk}) && (27.1)\\
&= \sum_{\mathbf{h}} \prod_r \sum_{v_r \in \{0,1\}} \exp(\sum_k v_r h_k W_{rk}) && (27.2)\\
&= \sum_{\mathbf{h}} \prod_r \left(1 + \exp((\sum_k h_k W_{rk}) \right) && (27.3)
\end{aligned}
$$

Chapter 28

Deep learning

Bibliography

Bernardo, J., and A. Smith. 1994. *Bayesian Theory*. John Wiley.

Bishop, C. 2006. *Pattern recognition and machine learning*. Springer.

Ding, Y., and R. Harrison. 2010. A sparse multinomial probit model for classification. *Pattern Analysis and Applications* pp. 1–9.

Figueiredo, M. 2003. Adaptive sparseness for supervised learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25:1150–1159.

Figueiredo, M., R. Nowak, and S. Wright. 2007. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE. J. on Selected Topics in Signal Processing*.

Ghahramani, Z., and G. Hinton. 1996. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, Dept. Comp. Sci., Univ. Toronto.

Jaakkola, T.S., and M.I. Jordan. 1999. Variational probabilistic inference and the QMR-DT network. *J. of AI Research* 10:291–322.

Jelinek, F. 1997. *Statistical methods for speech recognition*. MIT Press.

Lo, C. H. 2009. *Statistical methods for high throughput genomics*. PhD thesis, UBC.

MacKay, D. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Madigan, D., and A. Raftery. 1994. Model selection and accounting for model uncertainty in graphical models using Occam's window. *J. of the Am. Stat. Assoc.* 89:1535–1546.

Minka, T. 1998. Nuances of probability theory. Technical report, MIT Media Lab.

Rabiner, L. R. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. of the IEEE* 77:257–286.

Rogers, S., and M. Girolami. 2011. *A First Course in Machine Learning*. CRC Press.

Tanner, M. 1996. *Tools for statistical inference*. Springer.

Tipping, M. 1998. Probabilistic visualization of high-dimensional binary data. In *NIPS*.

Wainwright, M. J., and M. I. Jordan. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* 1–2:1–305.