CS 229 - Machine Learning
Xiangliang Zhang
Computer Science (CS)/Statistics (STAT) Program
Computer, Electrical and Mathematical Sciences & Engineering (CEMSE) Division
King Abdullah University of Science and Technology (KAUST)

HOMEWORK
II

Henrique Aparecido Laureano

Spring Semester 2018

Contents

Figure 1 gives an illustration of *sequential Bayesian learning* of a simple linear model of the form $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1\mathbf{x}$.

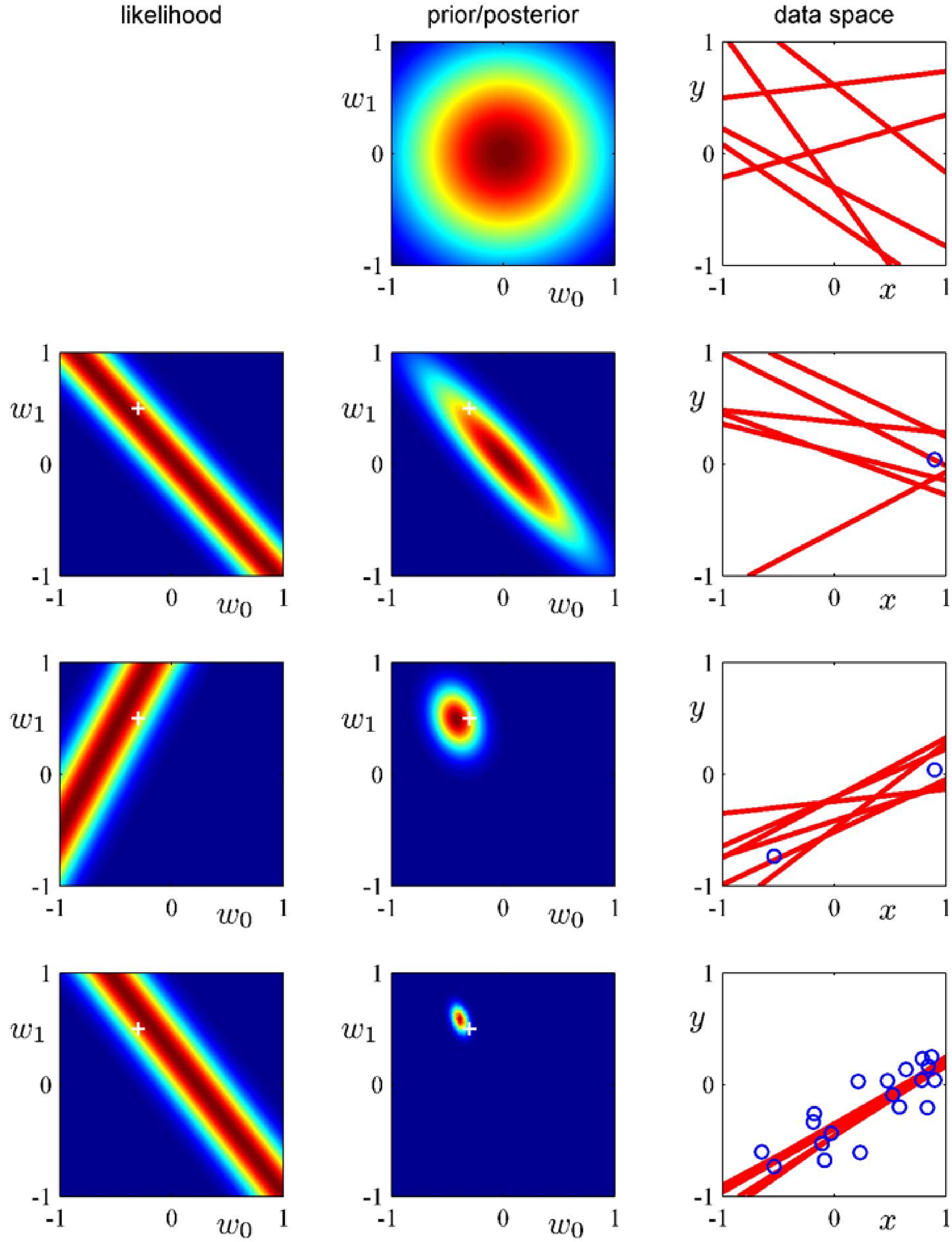

Figure 1: Illustration of sequential Bayesian learning of a simple linear model of the form $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1\mathbf{x}$.

(1)

Design a set of data samples in the linear model, with random noise.

Solution:

$$\begin{aligned} \mathbf{t} &= w_0 + w_1\mathbf{x} + \mathbf{noise} \\ \mathbf{x} &\sim \text{Uniform}(-1, 1) \\ \mathbf{noise} &\sim \text{Normal}(0, 0.2^2) \\ \mathbf{w} &= [w_0 \quad w_1]^\top = [-0.3 \quad 0.5]^\top \end{aligned}$$

□

(2)

Implement *sequential Bayesian learning*; show the results of *likelihood, prior/posterior, and examples in data space* in the same way as Figure 1.

Solution:

(*The results, graphs, are show in the end*)

$$\begin{aligned} \text{Prior}: \quad p(\mathbf{w}|\alpha) &\sim \text{Normal}(\mathbf{0}, \alpha^{-1}\mathbf{I}) \\ \alpha &= 2 \end{aligned}$$

Plotting the prior (for now no data, so this is equivalent to the posterior):

```
# <r code> =================================================================== #
alpha <- 2

prior <- Vectorize(                              # computing the prior normal density
  FUN = function(w0, w1) {                       # two "observations" w_{0} and w_{1}
    theta = list(w0 = w1, w1 = w1)
    w <- matrix(c(w0, w1), ncol = 1)
    exp( 1/alpha * t(w) %*% (1/alpha * diag(1, 2)) %*% w )
  },
  c("w0", "w1")
)
w0 <- seq(-1, 1, length.out = 150)             # grid of 150 values between -1 and 1
w1 <- seq(-1, 1, length.out = 150)             # grid of 150 values between -1 and 1
```

```
prior.w <- outer(w0, w1, prior)                      # applying the grids in the prior

par(mfrow = c(4, 3), mar = c(4, 4, 2, 1) + .1) ; plot.new()       # graphical setup
                                                                   # plotting the prior
image(w0, w1, prior.w, asp = 1, main = "prior/posterior", col = topo.colors(15)
      , xlab = expression(w[0]), ylab = expression(w[1])
      , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2))
contour(w0, w1, prior.w, col = "#0080ff", drawlabels = FALSE, add = TRUE)
# </r code> ================================================================== #
```

Six samples from the prior (posterior):

```
# <r code> =================================================================== #
library(MASS)             # function mvrnorm: to sample from a multivariate normal
sampling <- mvrnorm(6, c(0, 0), 1/alpha * diag(1, 2))       # each line is a sample
                  # plotting the samples (each sample (two points) describe a line)
plot(NA, xlim = c(-1, 1), ylim = c(-1, 1), xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2)
     , xlab = "x", ylab = "y", main = "data space")
for (i in 1:6) abline(sampling[i, ], col = 2, lwd = 3)
# </r code> ================================================================== #
```

$$\text{Likelihood}: \quad p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) \sim \text{Normal}(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1})$$
$$\propto \exp\Big(- \frac{\beta}{2} (\mathbf{t} - \mathbf{\Phi}\mathbf{w})^\top (\mathbf{t} - \mathbf{\Phi}\mathbf{w}) \Big)$$
$$\beta = (1/0.2)^2 = 25$$

Generating data and plotting the likelihood
(the white cross represent the real - used - coefficients):

```
# <r code> =================================================================== #
                  # generating from a uniform distribution of parameters -1 and 1
x1 <- runif(1, -1, 1)
noise <- rnorm(1, 0, .2)        # normal with mean zero and standard deviation 0.2
t1 <- -.3 + .5 * x1 + noise                          # w_{0}: -0.3 and w_{1}: 0.5

beta <- 25

like <- Vectorize(                                   # computing the likelihood density
  FUN = function(t, x, w0, w1) {
    theta = list(w0 = w1, w1 = w1)
    w = matrix(c(w0, w1), ncol = 1)
                                    #          building the model matrix (1 in the
    phi = matrix(c(1, x), ncol = 2) # first column corresponding to the intercept)
    math = t - phi %*% w
    exp( - beta/2 * t(math) %*% (math) )
  },
  c("w0", "w1")
```

```
)                                              # applying the grids in the likelihood
like.w <- outer(w0, w1, like, t = t1, x = x1)
                                                           # plotting the likelihood
image(w0, w1, like.w, asp = 1, main = "likelihood", col = topo.colors(15)
      , xlab = expression(w[0]), ylab = expression(w[1])
      , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2))
points(-.3, .5, col = "white", pch = 3, lwd = 3)                      # white cross
# </r code> ==================================================================== #
```

$$
\begin{aligned}
\text{Posterior}: \quad p(\mathbf{w}|\mathbf{t}) &\sim \text{Normal}(\mathbf{m}_N, \mathbf{S}_N) \\
&\propto \exp\Big(-\frac{\beta}{2}(\mathbf{t} - \mathbf{\Phi w})^\top(\mathbf{t} - \mathbf{\Phi w})\Big) \times \exp\Big(-\frac{1}{2}(\mathbf{w} - \mathbf{0})^\top \alpha^{-1}\mathbf{I}(\mathbf{w} - \mathbf{0})\Big) \\
\mathbf{m}_N &= \beta \mathbf{S}_N \mathbf{\Phi}^\top \mathbf{t} \\
\mathbf{S}_N^{-1} &= \alpha \mathbf{I} + \beta \mathbf{\Phi}^\top \mathbf{\Phi}
\end{aligned}
$$

Computing and plotting the posterior
(the white cross represent the real - used - coefficients):

```
# <r code> ===================================================================== #
post <- Vectorize(                                 # computing the posterior density
  FUN = function(t, x, w0, w1) {
    theta = list(w0 = w1, w1 = w1)
    w = matrix(c(w0, w1), ncol = 1)
    phi = matrix(c(rep(1, length(x)), x), ncol = 2)    # building the model matrix
    math = t - phi %*% w
    - beta/2 * t(math) %*% (math) - alpha/2 * t(w) %*% w
  },
  c("w0", "w1")
)                                               # applying the grids in the posterior
post.w <- outer(w0, w1, post, t = t1, x = x1)

image(w0, w1, post.w, asp = 1, col = topo.colors(15)      # plotting the posterior
      , xlab = expression(w[0]), ylab = expression(w[1])
      , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2))
points(-.3, .5, col = "white", pch = 3, lwd = 3)                      # white cross
contour(w0, w1, post.w                                     # plotting the contours
        , col = "#0080ff", drawlabels = FALSE, add = TRUE, nlevels = 18)
# </r code> ==================================================================== #
```

Six samples from the posterior:

```
# <r code> ===================================================================== #
sampling <- function(x, t) {                        # sampling from the posterior
  phi = matrix(c(rep(1, length(x)), x), ncol = 2)                          # \Phi
  sn = solve( alpha * diag(1, 2) + beta * t(phi) %*% phi )                 # S_{n}
  mn = beta * sn %*% t(phi) %*% t                                          # m_{N}
  mvrnorm(6, mn, sn)
```

```
}                             # sampling (each sample (two points) describe a line)
samps <- sampling(x = x1, t = t1)
plot(NA, xlim = c(-1, 1), ylim = c(-1, 1)                      # plotting the samples
     , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2), xlab = "x", ylab = "y")
for (i in 1:6) abline(samps[i, ], col = 2, lwd = 3)
points(x1, t1, col = "#0080ff", lwd = 3)                                 # data point
# </r code> ================================================================== #
```

Generating another data and plotting the likelihood:

```
# <r code> =================================================================== #
x2 <- runif(1, -1, 1) ; t2 <- -.3 + .5 * x2 + noise      # generating another data

like.w <- outer(w0, w1, like, t = t2, x = x2)                 # applying the grids
                                                           # plotting the likelihood
image(w0, w1, like.w, asp = 1, main = "likelihood", col = topo.colors(15)
      , xlab = expression(w[0]), ylab = expression(w[1])
      , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2))
points(-.3, .5, col = "white", pch = 3, lwd = 3)                        # white cross
# </r code> ================================================================== #
```

Computing and plotting the posterior:

```
# <r code> =================================================================== #
post.w <- outer(w0, w1, post, t = c(t1, t2), x = c(x1, x2))   # applying the grids

image(w0, w1, post.w, asp = 1, col = topo.colors(15)      # plotting the posterior
      , xlab = expression(w[0]), ylab = expression(w[1])
      , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2))
points(-.3, .5, col = "white", pch = 3, lwd = 3)                        # white cross
contour(w0, w1, post.w                                           # plotting contours
        , col = "#0080ff", drawlabels = FALSE, add = TRUE, nlevels = 18)
# </r code> ================================================================== #
```

More six samples from the posterior:

```
# <r code> =================================================================== #
samps <- sampling(x = c(x1, x2), t = c(t1, t2))                           # sampling

plot(NA, xlim = c(-1, 1), ylim = c(-1, 1)                      # plotting the samples
     , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2), xlab = "x", ylab = "y")
for (i in 1:6) abline(samps[i, ], col = 2, lwd = 3)
points(c(x1, x2), c(t1, t2), col = "#0080ff", lwd = 3)                  # data points
# </r code> ================================================================== #
```

Generating 20 data points and showing the likelihood for the last one:

```r
# <r code> =================================================================== #
x <- c(x1, x2, runif(18, -1, 1))                                # generating data
t <- c(t1, t2, -.3 + .5 * x[3:20] + noise)                     # generating data

like.w <- outer(w0, w1, like, t = t[20], x = x[20])          # applying the grids
                                                         # plotting the likelihood
image(w0, w1, like.w, asp = 1, main = "likelihood", col = topo.colors(15)
      , xlab = expression(w[0]), ylab = expression(w[1])
      , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2))
points(-.3, .5, col = "white", pch = 3, lwd = 3)                    # white cross
# </r code> ================================================================== #
```

Computing and plotting the posterior:

```r
# <r code> =================================================================== #
post.w <- outer(w0, w1, post, t = t, x = x)                  # applying the grids

image(w0, w1, post.w, asp = 1, col = topo.colors(15)      # plotting the posterior
      , xlab = expression(w[0]), ylab = expression(w[1])
      , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2))
points(-.3, .5, col = "white", pch = 3, lwd = 3)                    # white cross
contour(w0, w1, post.w                                         # plotting contours
        , col = "#0080ff", drawlabels = FALSE, add = TRUE, nlevels = 18)
# </r code> ================================================================== #
```

Six samples from the posterior:

```r
# <r code> =================================================================== #
samps <- sampling(x = c(x1, x2), t = c(t1, t2))                        # sampling

plot(NA, xlim = c(-1, 1), ylim = c(-1, 1)                  # plotting the samples
     , xaxp = c(-1, 1, 2), yaxp = c(-1, 1, 2), xlab = "x", ylab = "y")
for (i in 1:6) abline(samps[i, ], col = 2, lwd = 3)
points(x, t, col = "#0080ff", lwd = 3)                              # data points
# </r code> ================================================================== #
```

⇒

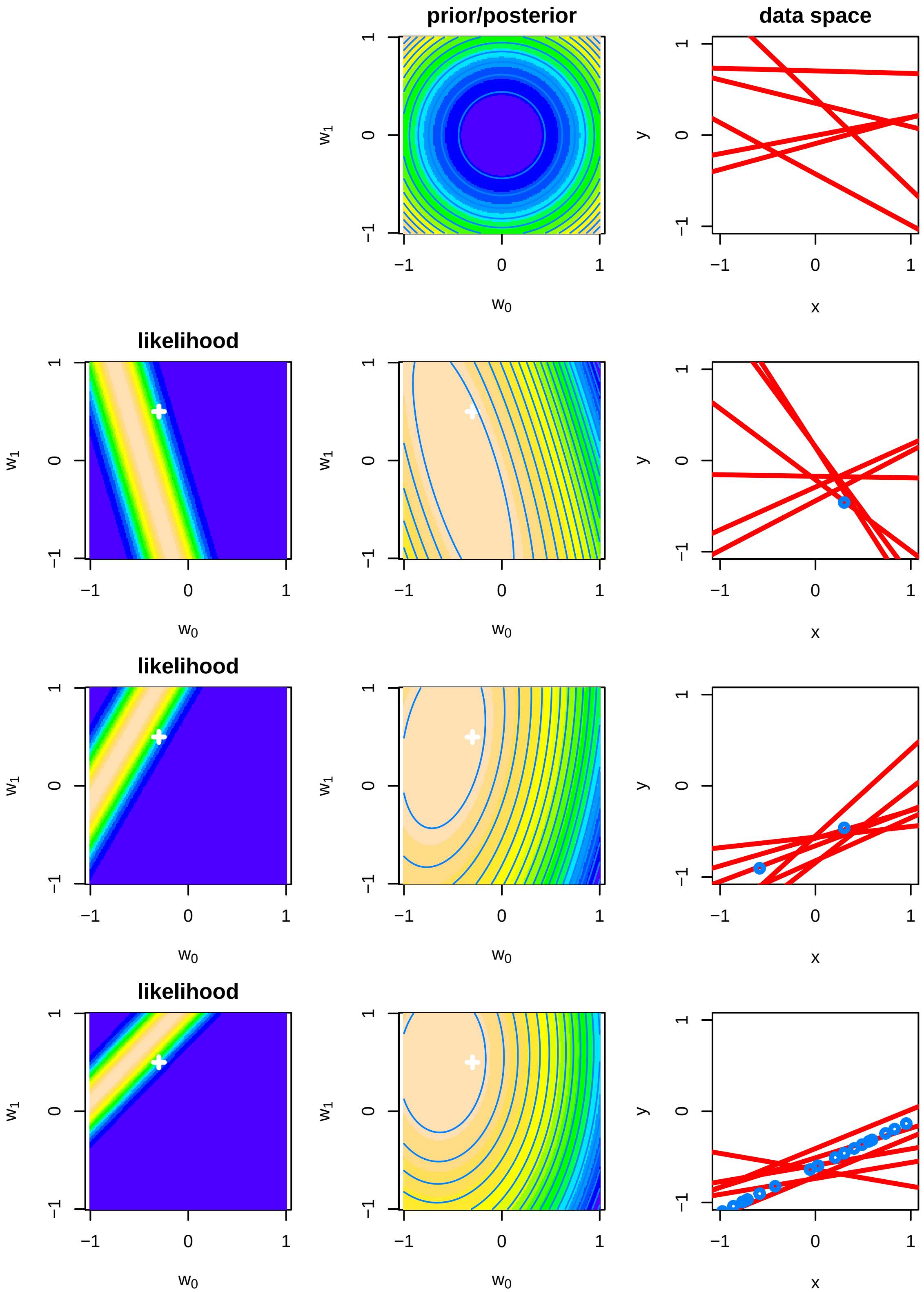

Figure 2: Illustration of sequential Bayesian learning of a simple linear model of the form $y(\mathbf{x}, \mathbf{w}) = w_0 + w_1\mathbf{x}$.

■