Department of Mathematics and Computer Science
University of Southern Denmark, Odense

April 25, 2011
Marco Chiarandini

DM825 - Introduction to Machine Learning

Sheet 3 – Solutions, Spring 2011 [pdf format]

Exercise 1 Bayesian prediction.

(a) Let $\theta \sim \mathsf{Dir}(\boldsymbol{\alpha})$. Consider multinomial random variables $(X_1, X_2, \ldots, X_N)$, where $X_n \sim \mathsf{Mult}(\theta)$ for each n, and where the X_n are assumed conditionally independent given θ. Now consider a random variable $X_{new} \sim \mathsf{Mult}(\theta)$ that is assumed conditionally independent of $(X_1, X_2, \ldots, X_N)$ given θ. Compute the predictive distribution:

$$p(x_{new}|x_1, x_2, \ldots, x_N, \alpha)$$

by integrating over θ.

Solution: The exercise refers to the theory developed in sec. 2.1 and 2.2 of [B1].

With multinomial distributions we consider the representation in which X_j is a random vector consisting of all 0's and a single 1. For example, $\vec{x} = (0,0,1,0,0,0)^T$. If we denote $p(x_k = 1) = \theta_k$ then $X_j \sim \mathsf{Mult}(\theta)$ corresponds to saying:

$$p(\vec{x}|\vec{\theta}) = \prod_{k=1}^{K} \theta_k^{x_k} \tag{1}$$

and $\vec{\theta} = (\theta_1, \ldots, \theta_K)^T$. This distribution is also known as generalized Bernoulli distribution.

Consequently, the likelihood for the training set $(X_1, X_2, \ldots, X_N)$ of independent observations is:

$$p(\vec{x}_1, \ldots, \vec{x}_N|\vec{\theta}) = \prod_{j=1}^{m}\prod_{k=1}^{K} \theta_k^{x_{jk}} = \prod_{k=1}^{K} \theta_k^{\sum_{j=1}^{m} x_{jk}} = \prod_{k=1}^{K} \theta_k^{l_k}$$

where we let l_k be the total number of x_j that belong to class k. The prior distribtuion of Θ is

$$\mathrm{Dir}(\vec{\theta}|\vec{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}$$

with $0 \leq \theta_k \leq 1$, $\sum_k \theta_k = 1$, $\vec{\alpha} = (\alpha_1, \ldots, \alpha_K)^T$ and $\alpha_0 = \sum_k \alpha_k$. The Dirichlet distribution is constructed with the aim of satisfying the conjugacy property. The fraction in front of the product is the normalizing coefficient derived from:

$$\frac{1}{g(\vec{\alpha})} \int \prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \mathrm{d}\vec{\theta} = 1 \tag{2}$$

$$g(\vec{\alpha}) = \frac{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)}{\Gamma(\alpha_0)} \tag{3}$$

The expected value for the kth component of the random variable $\vec{\Theta}$ is

$$\mathrm{E}[\Theta_k] = \frac{\alpha_k}{\alpha_0}.$$

From Bayes' Theorem

$$p(\vec{\theta}|\vec{x}_1,\ldots,\vec{x}_N) \propto p(\vec{x}_1,\ldots,\vec{x}_N|\vec{\theta})p(\vec{\theta}) \propto \prod_{k=1}^{K}\theta_k^{\alpha_k+l_k-1}$$

The posterior takes again the form of a Dirichlet distribution (conjugacy property) and comparing with the definition of the Dirichlet distribution above we can determine the normalization coefficients as

$$p(\vec{\theta}|\vec{x}_1,\ldots,\vec{x}_N) = \mathrm{Dir}(\vec{\theta}|\vec{\alpha}+\vec{l}) = \frac{\Gamma(\alpha_0+m)}{\Gamma(\alpha_1+l_1)\cdots\Gamma(\alpha_K+l_K)}\prod_{k=1}^{K}\theta_k^{\alpha_k+l_k-1} \tag{4}$$

with $\vec{l} = (l_1,\ldots,l_K)^T$.

To evaluate the predictive distribution of a new outcome we use the sum and product rules of probability

$$p(\vec{x}_{new}|\vec{x}_1,\ldots,\vec{x}_N,\alpha) = \int_{\vec{0}}^{\vec{1}} p(\vec{x}_{new}|\vec{\theta},\vec{\alpha})p(\vec{\theta}|\vec{x}_1,\ldots,\vec{x}_N,\vec{\alpha})\mathrm{d}\vec{\theta}$$

From (1) and (4) we have

$$\begin{aligned}
p(\vec{x}_{new}|\vec{x}_1,\ldots,\vec{x}_N,\alpha) &= \int p(\vec{x}_{new}|\vec{\theta},\vec{\alpha})p(\vec{\theta}|\vec{x}_1,\ldots,\vec{x}_N,\vec{\alpha})\mathrm{d}\vec{\theta} \\
&= \int\prod_{k=1}^{K}\theta_k^{x_{new,k}}\frac{1}{g(\vec{\alpha}+\vec{l})}\prod_{k=1}^{K}\theta_k^{\alpha_k+l_k-1}\mathrm{d}\vec{\theta} \\
&= \frac{1}{g(\vec{\alpha}+\vec{l})}\int\prod_{k=1}^{K}\theta_k^{\alpha_k+l_k+x_{new,k}-1} \\
&= \frac{g(\vec{\alpha}+\vec{x}_{new}+\vec{l})}{g(\vec{\alpha}+\vec{l})} \\
&= \frac{\Gamma(\alpha_0+m+1)\Gamma(\alpha_1+l_1)\cdots\Gamma(\alpha_K+l_K)}{\Gamma(\alpha_1+l_1+x_1)\cdots\Gamma(\alpha_K+l_K+x_k)\Gamma(\alpha_0+m)} \\
&= \frac{\alpha_k+l_k}{\alpha_0+m}
\end{aligned}$$

where k is chosen such that $x_{new,k} = 1$.

(b) Redo the problem in part (a), replacing the multinomial distribution with an arbitrary exponential family distribution, and the Dirichlet distribution with the corresponding exponential family conjugate distribution. You are to show that in general the predictive probability $p(x_{new}|x_1,x_2,\ldots,x_N)$ is a ratio of normalizers.

Solution: The exercise refers to the theory developed in sec. 2.4 and 2.4.2 of [B1]. Here we use a slightly different notation.

We first write out the likelihood for an arbitrary exponential family to find the form of the conjugate prior.

$$\begin{aligned}
p(x_1,\ldots,x_N|\eta) &= \left(\prod_j h(x_j)\right) g(\eta)^m \exp\left(\eta^T\sum_j u(x_j)\right) \\
&= \left(\prod_j h(x_j)\right) \exp\left(\eta^T\sum_j T(x_j) - mA(\eta)\right)
\end{aligned}$$

where we rewrote the exponential distribution in slightly different terms than we saw at lecture with $\exp\{-mA(\eta)\} = g(\eta)^m$ and $T = u$.

The conjugate family of priors has the same "form" as the likelihood to ensure that the posterior remains in the family of priors. Thus, for conjugate prior we use

$$p(\eta|\tau, n_0) = \frac{1}{Z(\tau, n_0)} \exp\left(\eta^T \tau - n_0 A(\eta)\right)$$

where $Z(\tau, n_0)$ is a normalizing function

$$Z(\tau, \eta_0) \stackrel{\text{def}}{=} \int \exp\left(\eta^T \tau - \eta_0 A(\eta)\right) d\eta$$

Then,

$$\begin{aligned} p(x_1, \ldots, x_m|\tau, n_0) &= \int p(x_1, \ldots, x_m|\eta) p(\eta|\tau) d\eta \\ &= \int \left(\prod_{j=1}^m h(x_j)\right) \exp\left(\eta^T \left(\tau + \sum_{j=1}^m T(x_j)\right) - (m + n_0) A(\eta)\right) d\eta \\ &= \left(\prod_{j=1}^m h(x_j)\right) Z\left(\tau + \sum_{j=1}^m T(x_j), m + n_0\right) \end{aligned}$$

Similarly

$$p(x_{new}, x_1, \ldots, x_m|\tau, n_0) = \left(h(x_{new}) \prod_{j=1}^m h(x_j)\right) Z\left(\tau + T(x_{new}) + \sum_{j=1}^m T(x_j), m + n_0 + 1\right)$$

The predictive probability is then, from product rule,

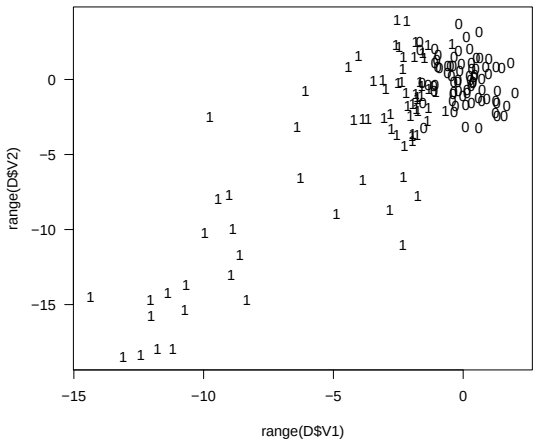
$$\begin{aligned} p(x_{new}|x_1, \ldots, x_m, \tau) &= \frac{p(x_{new}, x_1, \ldots, x_m|\tau)}{p(x_1, \ldots, x_m|\tau)} \\ &= \frac{\left(h(x_{new}) \prod_{j=1}^m h(x_j)\right) Z\left(\tau + T(x_{new}) + \sum_{j=1}^m T(x_j), m + n_0 + 1\right)}{\left(\prod_{j=1}^m h(x_j)\right) Z\left(\tau + \sum_{j=1}^m T(x_j), m + n_0\right)} \\ &= h(x_{new}) \frac{Z\left(\tau + T(x_{new}) + \sum_{j=1}^m T(x_j), m + n_0 + 1\right)}{Z\left(\tau + \sum_{j=1}^m T(x_j), m + n_0\right)} \end{aligned}$$

Exercise 2 Classification. The course website contains a data set of (x_n, y_n) pairs, where the x_n are 2-dimensional vectors and y_n is a binary label.

(a) Plot the data, using 0's and X's for the two classes. The plots in the following parts should be plotted on top of this plot.

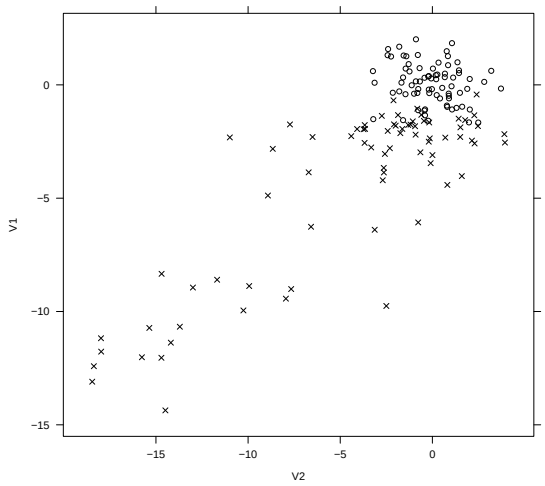
Solution

```
> D <- read.table("classification.dat")
> #plot(min(D$V1):max(D$V1),min(D$V1):max(D$V1),type="n")
> plot(range(D$V1),range(D$V2),type="n")
> text(D$V1,D$V2,D$V3)
```



Alternatively, with the lattice package (always explore the possibilities of the new functions you encounter via `example`).

```
> require(lattice)
> print(
      xyplot(V1~V2,groups=V3,
             data=D,pch=c(1,4))
      )
```



(b) Write a program to fit a logistic regression model using stochastic gradient ascent. Plot the line where the logistic function is equal to 0.5. Compare this outcome with the result attained using the `glm` function in R (check example in `predict.glm`).

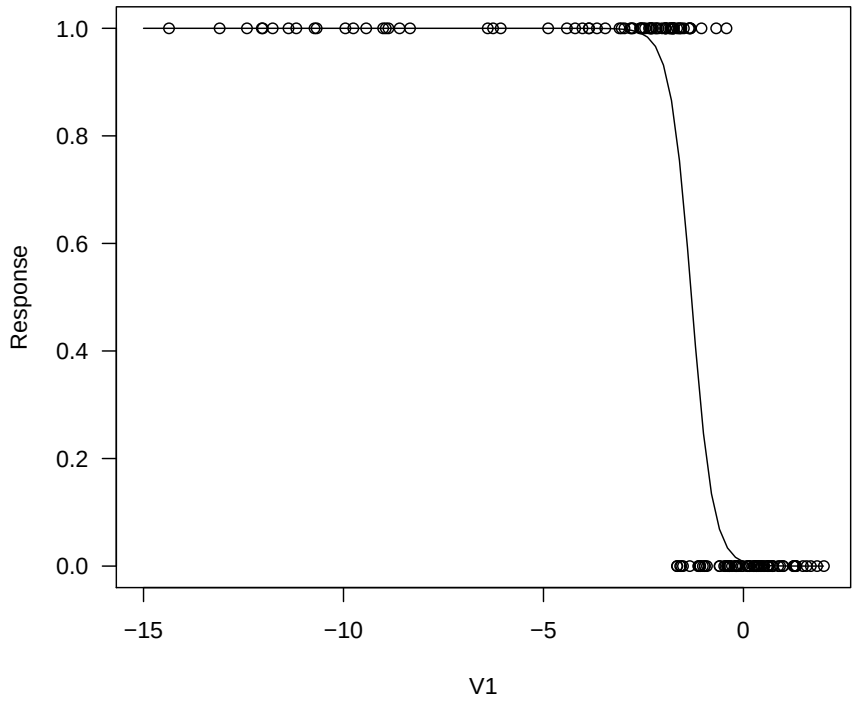
Solution

The line that corresponds to $p = 0.5$:

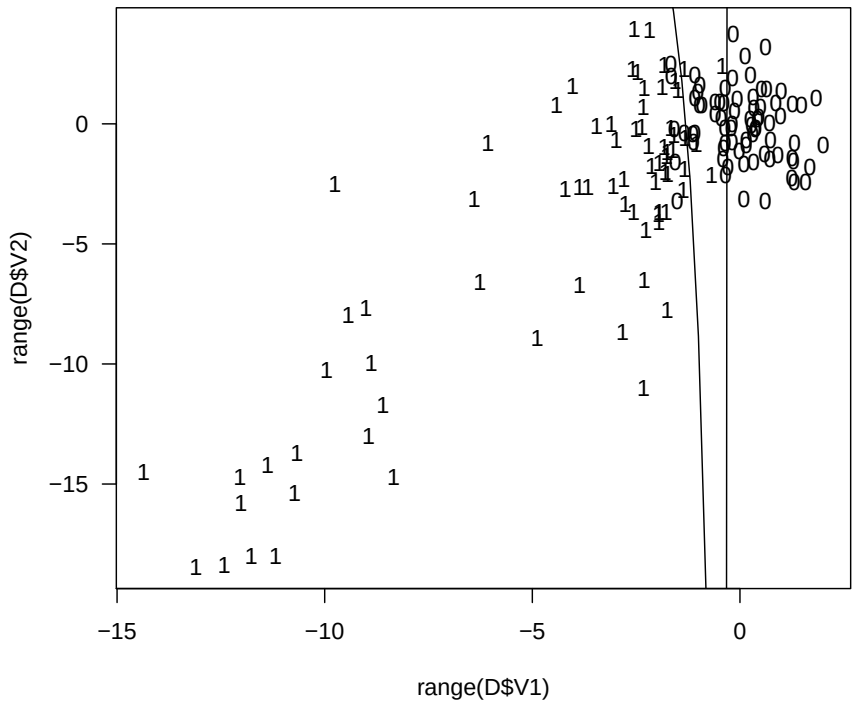
Let's investigate why we obtain two curves. Let's try to plot the linear discriminant in implicit form

Hence, in the previous plot the two lines where due to a discontinuity of the function that was linked by a line (ie, from $-\infty$ to ∞).

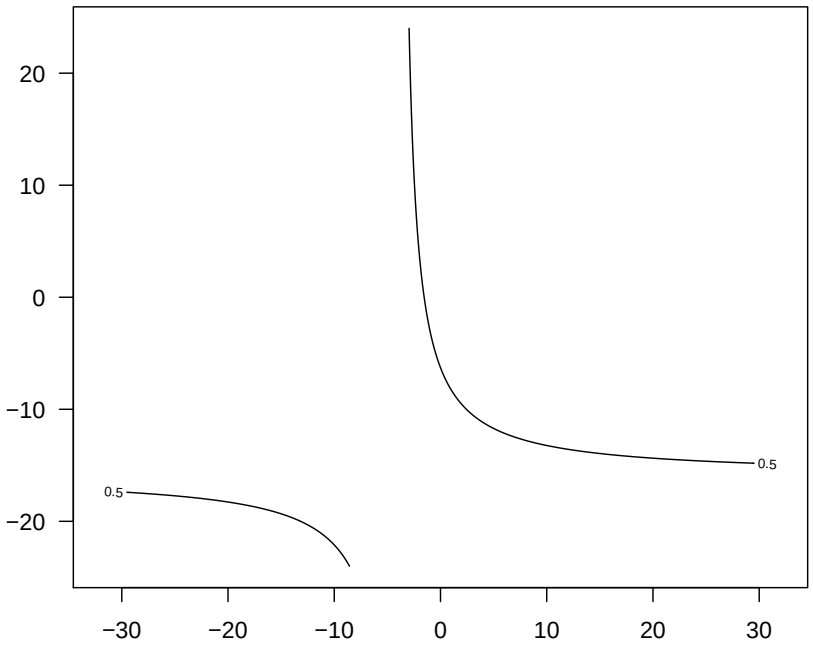
```
> reslogit <- glm(V3 ~ V1*V2, data=D, family=binomial(link="logit"))
> summary(reslogit)
> # components of the resulting object reslogit:
> # reslogit$coefficients: estimated regression coefficients
> # reslogit$fitted.values: estimated success probabilities
> # reslogit$residuals: residuals
> # reslogit$linear.predictors: the linear predictor b0+b1*x1+b2*x2
> #
>
>
> # Let's plot at a fixed value 0.8 for V2
>
> x1 <- seq(-15,2,.2)
> x2 <- -0.8
> lp <- reslogit$coefficients[1]+reslogit$coefficients[2]*x1+reslogit$coefficients[3]*x2+
> pr <- exp(lp)/(1+exp(lp))
> plot(D$V1,D$V3,xlim=c(-15,2),ylim=c(0,1),xlab="V1",ylab="Response")
> lines(x1,pr,lty=1)
```



```
> theta <- reslogit$coefficients
> plot(range(D$V1),range(D$V2),type="n")
> text(D$V1,D$V2,D$V3)
> matlines(x <- seq(-15,2,.2),(-theta[2]*x-theta[1])/(theta[3]+theta[4]*x),lwd=1)
```



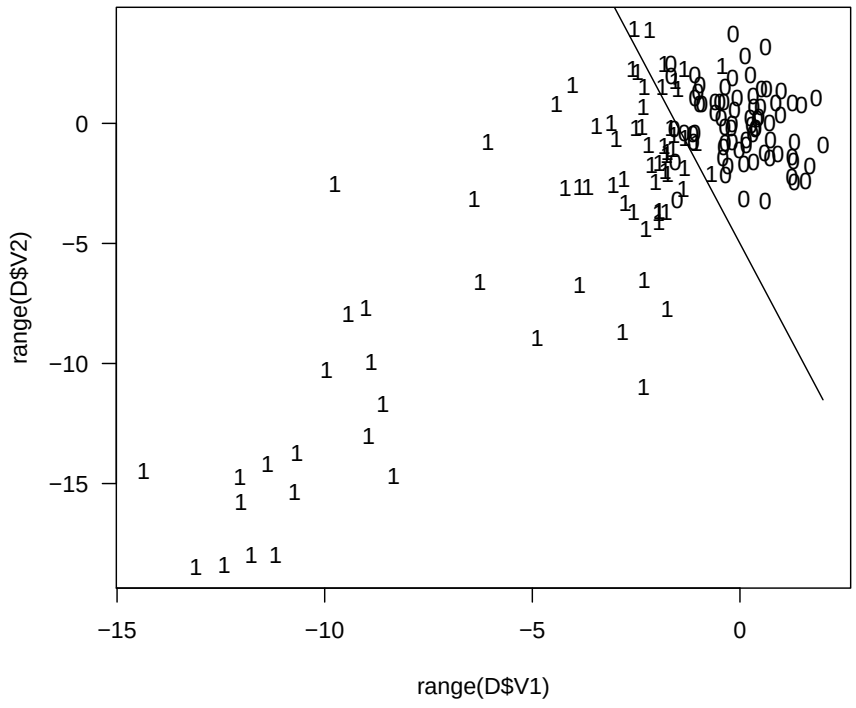
```
> x1 <- seq(-32,32,.2)
> x2 <- seq(-24,24,.2)
> f <- function(x,y) {
   apply(as.matrix(cbind(x,y)),1,
         function(l) theta%*%c(1,l[1],l[2],l[1]*l[2]))
 }
> zs <- outer(x1,x2,FUN=f)
> contour(x1,x2,zs,levels=0)
```



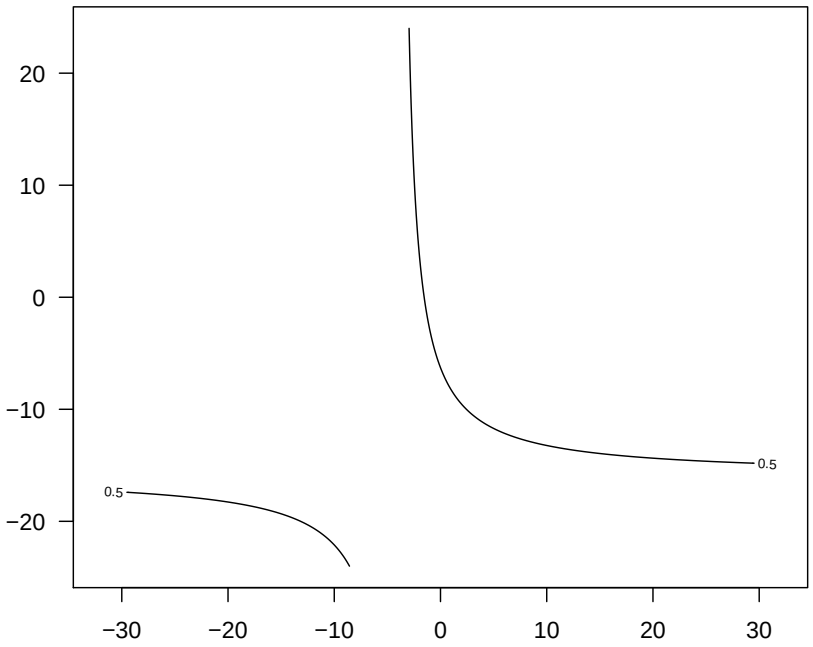
(c) Fit a linear regression to the problem, treating the class labels as real values 0 and 1. (You can solve the linear regression in any way you like, including solving the normal equations, using the LMS algorithm, or calling the built-in `lm` routine in R). Plot the line where the linear regression function is equal to 0.5.

Solution

```
> reslm <- lm(V3 ~ V1*V2, data=D)
> summary(reslm)
> theta <- reslm$coefficients
> plot(range(D$V1),range(D$V2),type="n")
> text(D$V1,D$V2,D$V3)
> matlines(x <- seq(-15,2,.2),(0.5-theta[2]*x-theta[1])/(theta[3]+theta[4]),lwd=1)
```



```
> x1 <- seq(-32,32,.2)
> x2 <- seq(-24,24,.2)
> f <- function(x,y) {
   apply(as.matrix(cbind(x,y)),1,
         function(l) theta%*%c(1,l[1],l[2],l[1]*l[2]))
 }
> zs <- outer(x1,x2,FUN=f)
> contour(x1,x2,zs,levels=0.5)
```



(d) The data set is a separate data set generated from the same source. Test your fits from parts (b), (c), and (d) on these data and compare the results.