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

June 17, 2011
Marco Chiarandini

DM825 - Introduction to Machine Learning

Sheet 7, Spring 2011 [pdf format]

Exercise 1 Neural Networks for Time Series Prediction.
A common data analysis task is time series prediction, where we have a set of data that show something varying over time, and we want to predict how the data will vary in the future. Examples are stock markets, river levels and house prices.
The data set PNoz.dat contains the daily measurement of the thickness of the ozone layer above Palmerston North in New Zealand between 1996 and 2004. Ozone thickness is measured in Dobson units, which are 0.01 mm thickness at 0 degree Celsius and 1 atmosphere pressure. The reduction in stratosferic ozone is partly responsible for global warming and the increased incidence of skin cancer. The thickness of the ozone varies naturally over the year, as you can see from the plot. (There are four fields in the data, and the ozone level is the third).

```
K <- read.table("PNoz.dat")
names(K) <- c("year","day","ozone.level","sulphur.dioxide.level")
plot(K$ozone.level,xlab="Time␣(Days)",ylab="Ozone␣(Dobson␣units)",pch=".",
    cex=1.5)
```

Your task is to use the multi-layer perceptron to predict the ozone levels into the future and see if you can detect an overall drop in the mean ozone level. Plot 400 predicted values together with the actual value.

The following is a reminder of the steps to carry out in the analysis:

- Select inputs and outputs for your problem and consequently the input and output nodes for the network.
- Normalize the data by rescaling.
- Split the data into training, validation and test (use the rule 50/25/25 if enough data or use cross validation with little data).
- Identify the main parameters to configure, e.g., the network architecture and others.
- Train the network and compare for different parameters
- Assess the performance on the test data.
- Analyse the bias and variance trade off.

Exercise 2 Probability theory.
In class we often used the rule:

$$p(x_i|x_{-i}) = \frac{p(x_1,\ldots,x_N)}{\int p(x_1,\ldots,x_N)dx_i}$$

where $x_{-i} = \{x_1,\ldots,x_N\} \setminus x_i$. Derive this rule from the product rule and sum rule.

Solution: By the product rule

$$p(x_1, \ldots, x_N) = p(x_i|x_{-i})p(x_{-i})$$

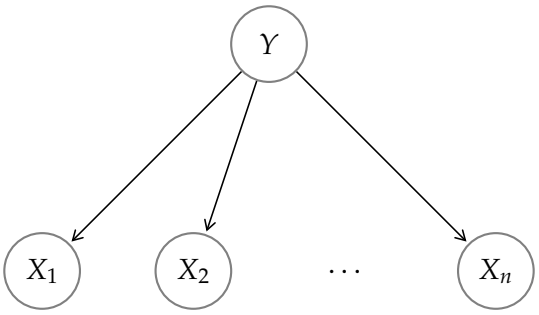
Rearranging and marginalizing:

$$\begin{aligned} p(x_i|x_{-i}) &= \frac{p(x_1, \ldots, x_N)}{p(x_{-i})} \\ &= \frac{p(x_1, \ldots, x_N)}{\int p(x_1, \ldots, x_N) dx_i} \end{aligned}$$

Exercise 3 Naive Bayes.
Consider the binary classification problem of spam email in which a binary label $Y \in \{0, 1\}$ is to be predicted from a feature vector $X = (X_1, X_2, \ldots, X_n)$, where $X_i = 1$ if the word i is present in the email and 0 otherwise. Consider a naive Bayes model, in which the components X_i are assumed mutually conditionally independent given the class label Y.

a Draw a directed graphical model corresponding to the naive Bayes model.

Solution:



b Find a mathematical expression for the posterior class probability $p(Y = 1|x)$, in terms of the prior class probability $p(Y = 1)$ and the class-conditional densities $p(x_i|y)$.

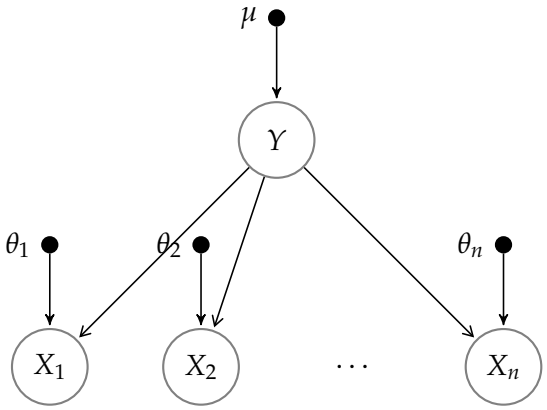
Solution:

$$\begin{aligned} p(Y = 1|x) &= \frac{p(\vec{x}|Y = 1)p(Y = 1)}{p(\vec{x})} \\ &= \frac{\Pi_{i=1}^{n} p(x_i|Y = 1)p(Y = 1)}{\sum_{y=0,1} \Pi_{i=1}^{n} p(x_i|Y = y)p(Y = y)} \end{aligned}$$

c Make now explicit the hyperparameters of the Bernoulli distributions for Y and X_i. Call them, μ and θ_i, respectively. Assume a beta distribution for the prior of these hyperparameters and show how to learn the hyperparameters from a set of training data $\mathbf{d} = (y^j, \vec{x}^j)_{j=1}^{m}$ using a Bayesian approach. Compare this solution with the one developed in class via maximum likelihood.

Solution

The hierarchical model is represented in the figure.



For Y we assume

$$p(Y = 1|\mu) = \text{Bern}(\mu) = \mu$$

For X_i we the distribution depends by the parent and we assume

$$p(X_i = 1|Y = 1, \theta_{i1}) = \text{Bern}(\theta_{i1}) = \theta_{i1}$$
$$p(X_i = 1|Y = 0, \theta_{i0}) = \text{Bern}(\theta_{i0}) = \theta_{i0}$$

The prior distribution on the θs and μ captures the uncertainty on these parameters. Assuming a *beta distribution* and referring by θ to both the θ_{iy}s and μ

$$p(\theta) = \text{Beta}(\theta|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

The Gamma function $\Gamma(\cdot)$ is a normalizing function. The parameters α and β with $\alpha > 0$ and $\beta > 0$ are *hyperparameters* of the prior distribution. The mean of a beta distribution is $E[\theta] = \frac{\alpha}{\alpha+\beta}$.

The beta distribution has the *conjugacy property*, that is, the posterior distribution has the same functional form as the prior. This property is convenient because the posterior can be derived in closed form. For the Y node:

$$\begin{aligned} p(\mu|\mathbf{d}) &= \frac{p(\mathbf{d}|\mu)p(\mu)}{p(\mathbf{d})} \\ &= \frac{\text{Bin}(s|\mu)p(\mu)}{p(\mathbf{d})} \\ &\propto \text{Beta}(\mu|\alpha + s, \beta + (m - s)). \end{aligned}$$

where s are the cases of m with $Y = 1$.

For the X_i nodes assuming the independence

$$p(\vec{\theta}|\mathbf{d}) = \Pi_{j=1}^{m}\Pi_{y=0,1}p(\theta_{jy}|\mathbf{d})$$

and

$$p(\theta_{i1}|d) = \text{Beta}(\mu|\alpha + s_{i1}, \beta + (s - s_{i1}))$$

where s_{i1} is the number of cases in $\mathbf{d}$ with $X_i = 1$ and $Y = 1$ and s is the number of cases in $\mathbf{d}$ with $Y = 1$.

Thus the prediction for each variable after learning occurred is given by

$$p(Y=1|\mathbf{d}) = \sum p(Y=1|\mathbf{d})p(\mu|\mathbf{d}) = E_{p(\mu|\mathbf{d})}[\mu|\mathbf{d}] = \frac{\alpha+s}{\alpha+\beta+m}$$

$$p(X_i=1|Y=1,\mathbf{d}) = \sum p(X_i=1|Y=1,\mathbf{d},\theta_{i1})p(\theta_{i1}|\mathbf{d}) = E_{p(\theta_{i1}|\mathbf{d})}[\theta_{i1}|\mathbf{d}] = \frac{\alpha+s_{i1}}{\alpha+\beta+s}$$

This is very similar to what we saw in class derived from the joint likelihood:

$$\begin{aligned}
\phi_y &= \frac{\sum_{j=1}^{m} I\{Y^j=1\}}{m} = \frac{s}{m} \\
\phi_{i|Y=1} &= \frac{\sum_{j=1}^{m} I\{X_i^j=1, Y^j=1\}}{\sum_{j=1}^{m} I\{Y^j=1\}} = \frac{s_{i1}}{s}
\end{aligned}$$

If we want to predict Y given $\vec{x}$ then we use:

$$\begin{aligned}
p(Y=1|x,\mathbf{d}) &= \frac{p(\vec{x}|Y=1,\mathbf{d})p(Y=1,\mathbf{d})}{p(\vec{x},\mathbf{d})} \\
&= \frac{\Pi_{i=1}^{n} p(x_i|Y=1,\mathbf{d})p(Y=1,\mathbf{d})}{\sum_{y=0,1}\Pi_{i=1}^{n} p(x_i|Y=y,\mathbf{d})p(Y=y,\mathbf{d})}
\end{aligned}$$