STAT 675 – Homework 3 (solutions)

Solutions compiled by: Zach Weller, Ryan Haunfelder, Aaron Nielsen

Due: Oct. 23

1. Let's look at the digits data with sparse logistic regression, LDA, and (kernel) support vector machines.

   (a) Download the digits data (`http://yann.lecun.com/exdb/mnist/`) and read the website for relevant information about the dataset.

   R code (load data):

```
install.packages("darch")
install.packages("e1071")
library(darch)
library(e1071)
library(MASS)
library(fields)
library(glmnet)
setwd("~/Downloads/")
readMNIST("~/Downloads/")
load("train.Rdata")
load("test.Rdata")
```

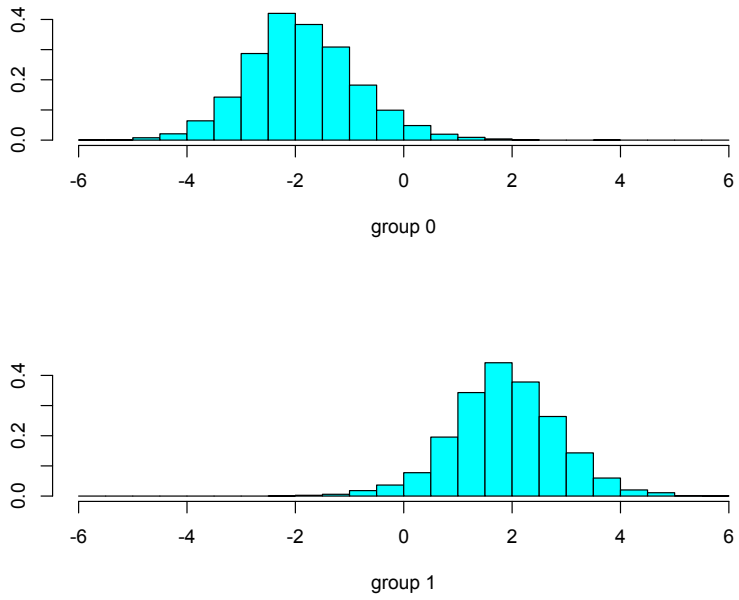
   See "DigitsScript.R" for a full list of R commands. WARNING: portions of the script are very computationally demanding and can take some time to run.

   (b) We wish to do two-class classification in two cases. Apply the above mentioned techniques to these data sets.

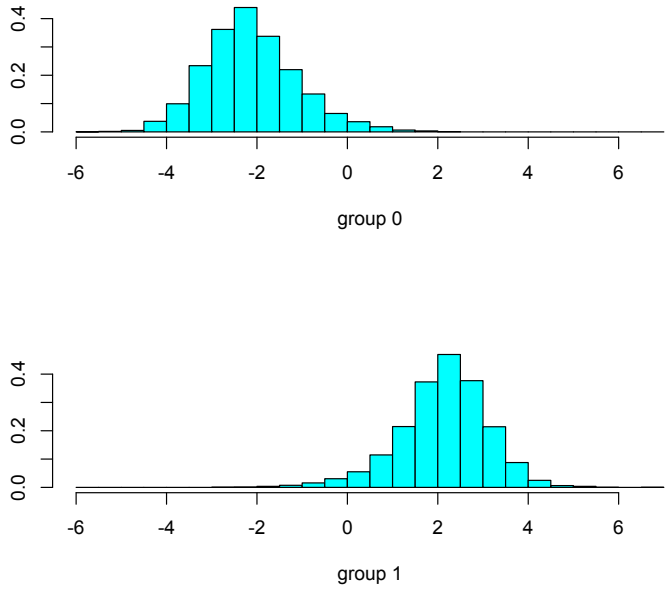
      i. Comparing 3's to 5's

      ii. Comparing 4's to 9's.

   Be sure to plot the results of LDA on the plane $\mathcal{H}_1$. Try different kernels for the SVM.
   Solution:



i.



ii.

(c) By looking at $\mathcal{H}_1$, do you feel a kernelization using these coordinates as data would improve the classification rate.

Solution:

Based on the two plots above, it does not appear that kernelization would improve the classification rates. It would likely be difficult to conceive a kernel that would separate the overlap between the two classes.

(d) Which of the methods has a better test error rate?

Solution:

Problem 1bi error rates:

	SpLR	LDA	SVM-pol	SVM-rad	SVM-sig
1	0.03155	0.03470	0.01472	0.00789	0.00789

SVM with either a radial or sigmoidal kernel performed the best.

Problem 1bii error rates:

	SpLR	LDA	SVM-pol	SVM-rad	SVM-sig
1	0.02562	0.03616	0.01507	0.01507	0.01507

SVM again performed the best. This time, the choice of kernel did not matter.

(e) State how you could kernelize LDA into a nonlinear method. Note: It is clear how to kernelize it through a feature map. This might be very expensive, however. Can you kernelize it through inner products instead?

Solution:

One method to kernalize LDA into a nonlinear model is outlined in Scholkopft (1999)[1]. Let Φ be a non-linear mapping into some feature space $\mathcal{F}$. To find the linear discriminant in $\mathcal{F}$ the following must be maximized: $J(w) = \frac{w^T S_B^\Phi w}{w^T S_W^\Phi w}$, where $w \in \mathcal{F}$ and S_B^Φ and S_W^Φ in $\mathcal{F}$. That is,

$$S_B^\Phi := (m_1^\Phi - m_2^\Phi)(m_1^\Phi - m_2^\Phi)^T$$
$$S_W^\Phi := \sum_{i=1,2} \sum_{m \in \mathcal{X}_i} (\Phi(x) - m_i^\Phi)$$
$$m_i^\Phi = \frac{1}{l_i} \sum_{j=1}^{l_i}$$

.

Solving this can be computational complex or even be impossible to solve directly. Instead, you can kernalize it through inner products using the "kernel trick." The dot product in the new feature space $\mathcal{F}$ can be replaced by a kernel function $k(x, y) = \phi(x)\phi(y)$. For more information on this method, refer to Scholkopft (1999).

2. Show the following (using notation from lecture)

(a)

$$\mathbb{P}(Y \neq \hat{g}(X)|X = x) = 1 - (\mathbf{1}m(x) + (1 - \mathbf{1})(1 - m(x))),$$

where $\mathbf{1} \equiv \mathbf{1}_{g(x)=1}(x)$.

Solution:

$$\begin{aligned}\mathbb{P}(Y \neq \hat{g}(X)|X = x) &= 1 - \mathbb{P}(Y = \hat{g}(X)|X = x) \\ &= 1 - \left(\mathbb{P}(Y = 1|X = x)\mathbf{1}_{\hat{g}(x)=1}(x) + \mathbb{P}(Y = 0|X = x)(1 - \mathbf{1}_{\hat{g}(x)=1}(x))\right) \\ &= 1 - \left(m(x)\mathbf{1}_{\hat{g}(x)=1}(x) + (1 - m(x))(1 - \mathbf{1}_{\hat{g}(x)=1}(x))\right)\end{aligned}$$

[1] Scholkopft, Bernhard, and Klaus-Robert Mullert. "Fisher discriminant analysis with kernels." Neural networks for signal processing IX (1999).

(b)

$$\mathbb{P}(Y \neq \hat{g}(X)|X = x) - \mathbb{P}(Y \neq g^*(X)|X = x) = 2|m(x) - 1/2|\mathbf{1}_{g^*(x)\neq\hat{g}(x)}(x)$$

Solution:

$$\begin{aligned}
&\mathbb{P}(Y \neq \hat{g}(X)|X = x) - \mathbb{P}(Y \neq g^*(X)|X = x) \\
&= (1 - (m(x)\mathbf{1} + (1 - \mathbf{1})(1 - m(x)))) - \big(1 - \big(m(x)\mathbf{1}_{g^*(x)=1}(x) + (1 - m(x))(1 - \mathbf{1}_{g^*(x)=1}(x))\big)\big) \\
&= m(x)(\mathbf{1}_{g^*(x)=1}(x) - \mathbf{1}_{\hat{g}(x)=1}(x)) + (\mathbf{1}_{\hat{g}(x)=1}(x) - \mathbf{1}_{g^*(x)=1}(x)) + m(x)(\mathbf{1}_{g^*(x)=1}(x) - \mathbf{1}_{\hat{g}(x)=1}(x)) \\
&= 2m(x)(\mathbf{1}_{g^*(x)=1}(x) - \mathbf{1}_{\hat{g}(x)=1}(x)) - (\mathbf{1}_{\hat{g}(x)=1}(x) - \mathbf{1}_{g^*(x)=1}(x)) \\
&= (2m(x) - 1)(\mathbf{1}_{g^*(x)=1}(x) - \mathbf{1}_{\hat{g}(x)=1}(x)) \\
&= \begin{cases} 2(m(x) - 1/2) & \text{if } g^*(x) = 1 \text{ and } \hat{g}(x) = 0 \\ 2(1/2 - m(x)) & \text{if } g^*(x) = 0 \text{ and } \hat{g}(x) = 1 \\ 0 & \text{otherwise} \end{cases} \\
&= 2|m(x) - 1/2|\mathbf{1}_{g^*(x)\neq\hat{g}(x)}(x)
\end{aligned}$$

(c) Use these results to show that g^* is the Bayes' rule.
Solution: For a squared error loss function, $l_c(Y) = \mathbb{E}_{Y|X=x}(Y - c)^2$,

$$\begin{aligned}
\mathbb{E}_{Y|X=x} l(\hat{g}(X), Y) &= \mathbb{E}_{Y|X=x}(Y - \hat{g}(X))^2 \\
&= P((Y - \hat{g}(X))^2 = 1|X = x) \\
&= P(Y \neq \hat{g}(X)|X = x)
\end{aligned}$$

and,

$$\begin{aligned}
\mathbb{E}_{Y|X=x} l(g^*(X), Y) &= \mathbb{E}_{Y|X=x}(Y - g^*(X))^2 \\
&= P((Y - g^*(X))^2 = 1|X = x) \\
&= P(Y \neq g^*(X)|X = x)
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}_{Y|X=x} l(g^*(X), Y) - \mathbb{E}_{Y|X=x} l(\hat{g}(X), Y) &= \mathbb{P}(Y \neq \hat{g}(X)|X = x) - \mathbb{P}(Y \neq g^*(X)|X = x) \\
&= 2|m(x) - 1/2|\mathbf{1}_{g^*(x)\neq\hat{g}(x)}(x) \\
&> 0
\end{aligned}$$

so the risk for any other estimator $\hat{g}(X)$ is greater than that of $g^*(X)$ and hence $g^*(X)$ is the Bayes' rule.