Naïve Bayes and KNN

Chapter 13: Nonlinear Classification Models

Naïve Bayes

- Naïve Bayes (NB) assumes that features are independent in each class. It is useful when the number of features p is large, and so multivariate methods like QDA and even LDA break down.
- NB can be easily used for qualitative predictors, for which, replace $f_{jk}(x_j)$ with probability mass function over discreate categories.
- Despite strong assumption of independence, NB often produces good classification results.

Logistic regression vs. LDA

- Similarity: Both logistic regression and LDA produce *linear* boundaries.
- Difference lies in fitting procedures
  - LDA assumes that the observations are drawn from the Gaussian distribution with a same variance in each class, while logistic regression does not have this assumption.
  - LDA would do better than logistic regression if the assumption of normality hold, otherwise, logistic regression could outperform LDA.

data preprocessing for the predictors to be symmetric → LDA

Logistic regression vs. LDA

- Similarity: Both logistic regression and LDA produce *linear* boundaries.
- Difference lies in fitting procedures

  $\frac{\mu_1 + \mu_2}{2} - X \gtrless 0$

  - LDA assumes that the observations are drawn from the Gaussian distribution with a same variance in each class, while logistic regression does not have this assumption.
  - LDA would do better than logistic regression if the assumption of normality hold, otherwise, logistic regression could outperform LDA.

KNN vs. LDA

- KNN is completely non-parametric: No assumptions are made about the shape of the decision boundary.
- Advantage of KNN: We can expect KNN to dominate LDA and logistic regression when the decision boundary is highly non-linear.
- Disadvantage of KNN: KNN does not tell us which predictors are important (no table of regression coefficients)

QDA vs. LDA, logistic regression, KNN

highly non-linear

- QDA is compromise between non-parametric KNN method and the linear LDA and logistic regression.

linear boundary

- If the true decision boundary is
  - Linear: LDA and logistic regression outperform;
  - Moderately non-linear: QDA outperforms;
  - More complicated (highly nonlinear): KNN is superior.
- Note that logistic regression could also fit quadratic boundaries, like QDA, by explicitly including quadratic terms in the model.

overfitting.

Summary

- Logistic regression is very popular for classification, especially when K =2 (binary classification)
- LDA is useful when the sample size *n* is small, or the classes are well separately, and Gaussian (normal) assumptions are reasonable. Also, when K >2, QDA requires large *n*.
- KNN is useful when the parametric methods do not work well.
- Naïve Bayes is useful when the number of predictors *p* is very large.

Naïve Bayes

```
##################################Naïve Bayes#####################################
set.seed(476)
NBTune <- train(x = as.matrix(Smarket.train[,1:8]),
y = Smarket.train$Direction,
method = "nb",
preProc = c('center', 'scale'),
metric = "ROC",
trControl = ctrl)
NBTune
```

Naïve Bayes output

```
Naive Bayes

998 samples
  8 predictor
  2 classes: 'Down', 'Up'

Pre-processing: centered (8), scaled (8)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 750, 750, 750, 750, 750, 750, ...
Resampling results across tuning parameters:

  usekernel  ROC        Sens       Spec
  FALSE      0.9958964  0.9275410  0.9885714
   TRUE      0.9952485  0.9714754  0.9657143

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was held constant at a value of 1
ROC was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = FALSE and adjust = 1.
```

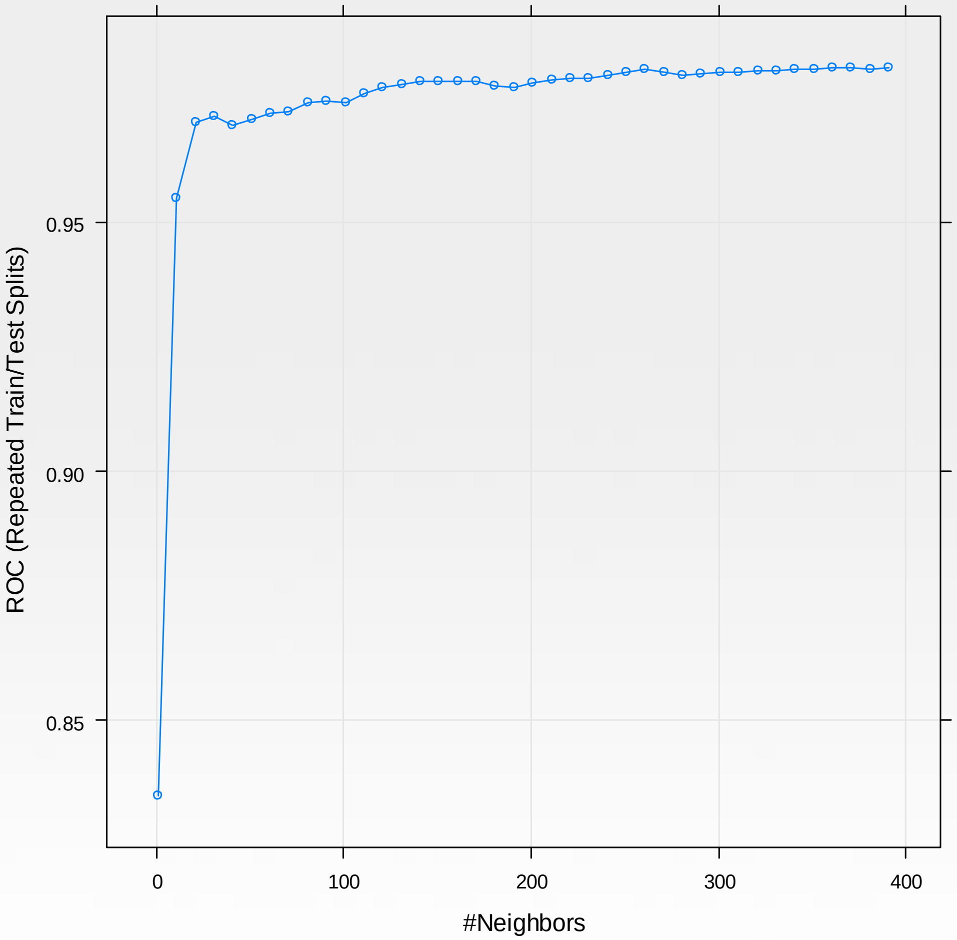
KNN

```
#################################K-nearest neighbors################################

 set.seed(1)
 KNNTune <- train(x = as.matrix(Smarket.train[,1:8]),
              y = Smarket.train$Direction,
 method = "knn",
 metric = "ROC",
 preProc = c("center", "scale"),
 tuneGrid = data.frame(.k =  seq(1,400, by=10)),
 trControl = ctrl)
 plot(KNNTune)
```

k = 1, 11, 21, . .

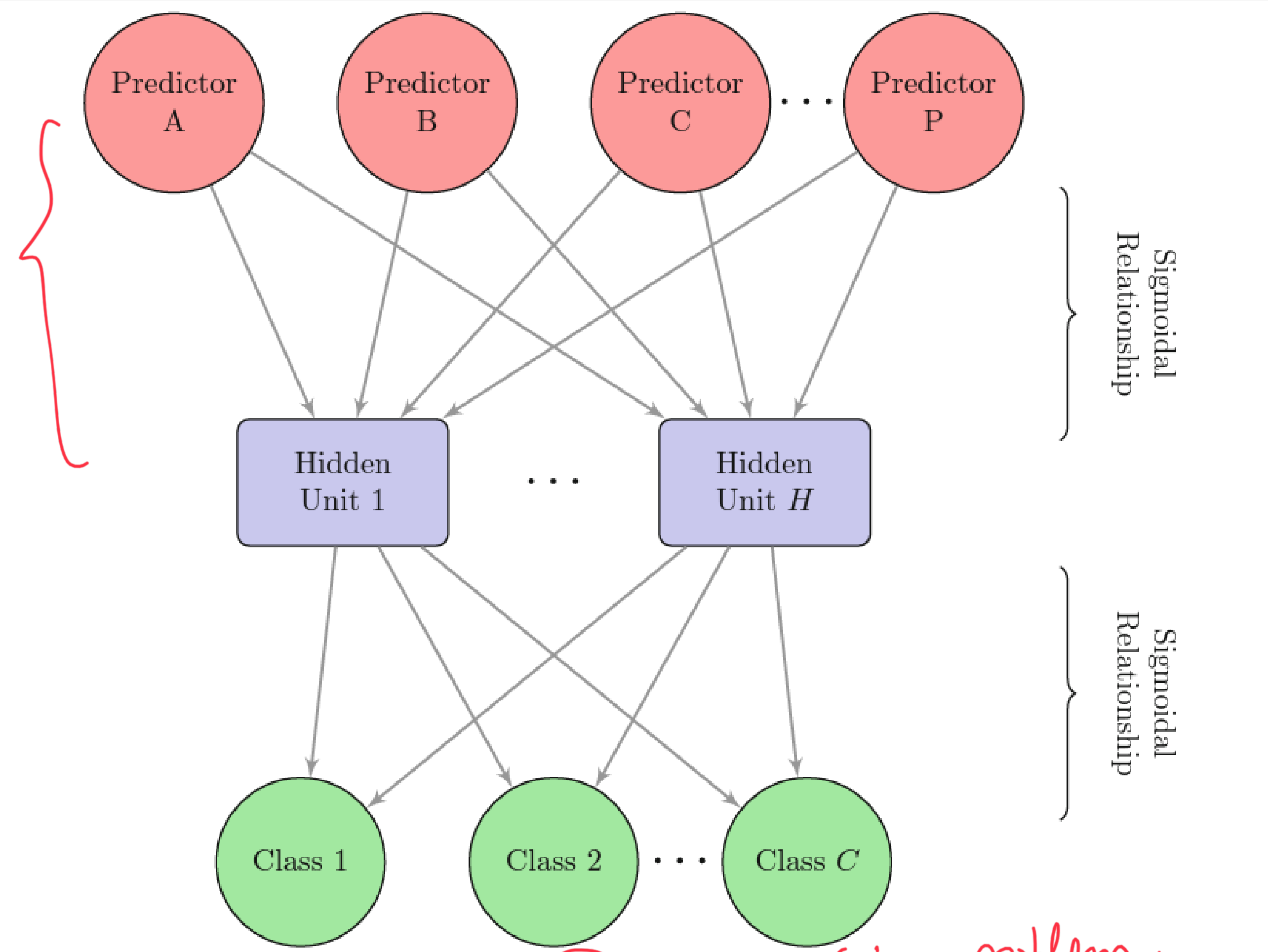
KNN output (K = 391)



Neural networks

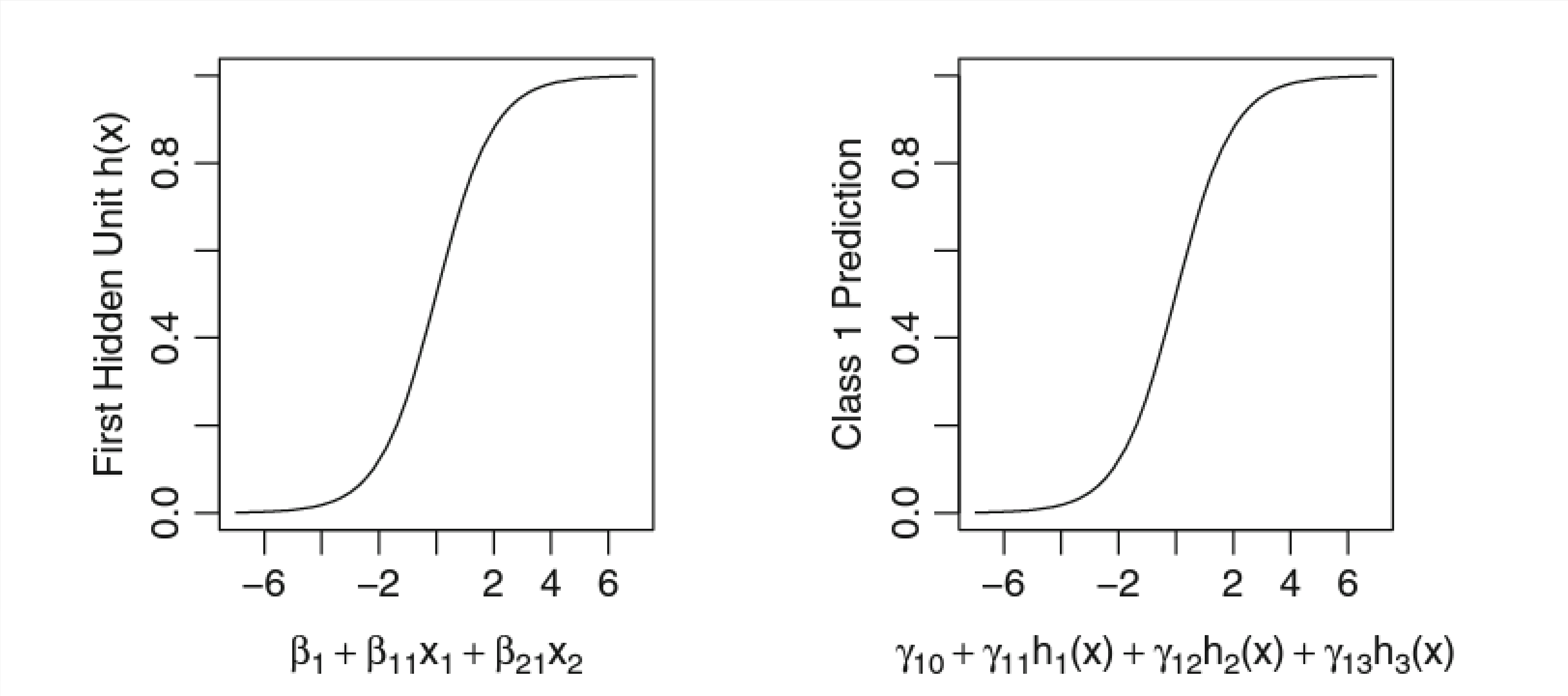
Chapter 13: Nonlinear Classification Models

Neural networks



regression problem

Neural networks



- A diagram of a neural network for classification with a single hidden layer.
- The hidden units are linear combinations of the predictors that have been transformed by a sigmoidal function.
- The output is also modeled by a sigmoidal function

Remarks on neural networks

- Like their regression counterparts, neural networks for classification have a significant potential for *over-fitting*. However, model averaging helps reduce over-fitting.
- Collinearity and non-informative predictors will have a comparable impact on model performance.
- To increase the effectiveness of neural networks, *various transformations of the data were evaluated*. One in particular, the spatial sign transformation, had a significant positive impact on the performance of the neural networks for these data.

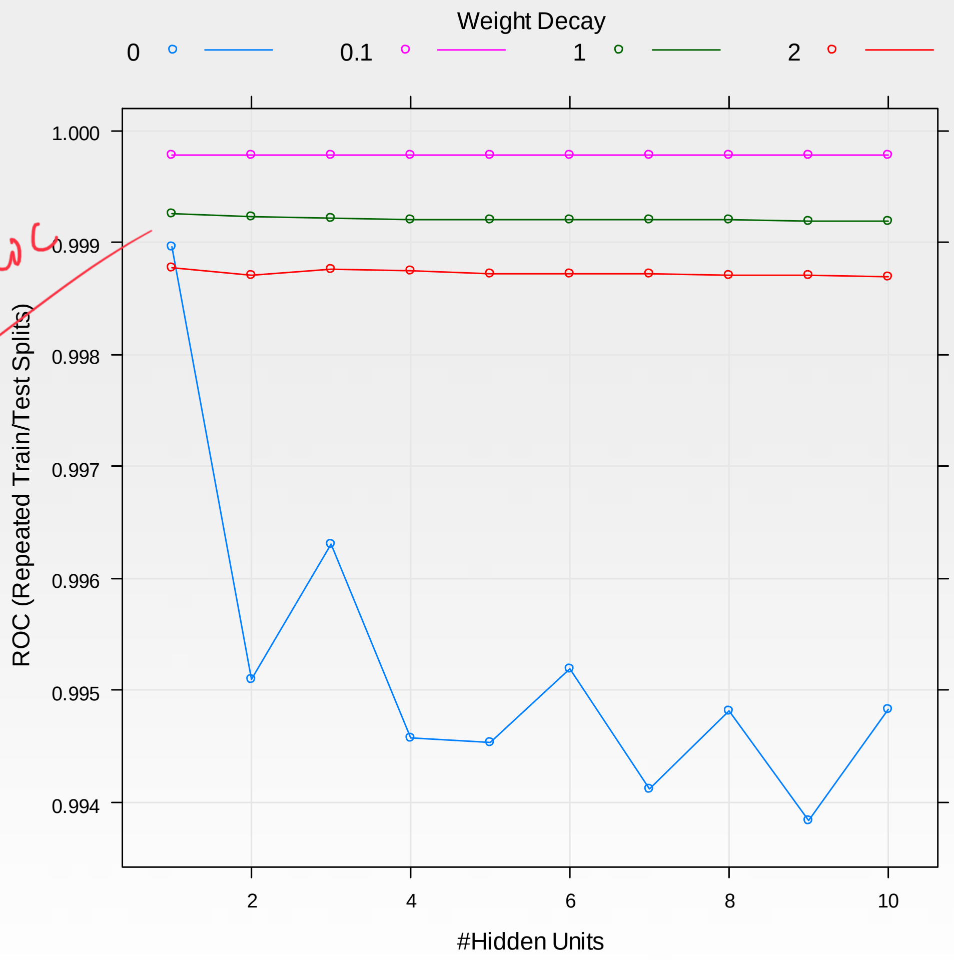
Neural networks

```
################################Neural networks################################
set.seed(476)
nnetGrid <- expand.grid(.size = 1:10,
.decay = c(0, .1, 1, 2))
maxSize <- max(nnetGrid$.size)
numWts <-200
NNTune <- train(x = as.matrix(Smarket.train[,1:8]),
             y = Smarket.train$Direction,
method = "nnet",
metric = "ROC",
preProc = c("center", "scale", "spatialSign"),
tuneGrid = nnetGrid,
trace = FALSE,
maxit = 2000,
MaxNWts = numWts,
trControl = ctrl)
NNTune
plot(NNTune)
```

hidden units.

Neural networks output

maximum Roc



Flexible Discriminant Analysis

Chapter 13: Nonlinear Classification Models

Flexible discriminant analysis (FDA)

- FDA allows the idea of linear discriminant analysis to be extended in a number of ways:
  - Many of the models in Chapters 6 and 7, such as the lasso, ridge regression, or MARS, can be extended to create discriminant variables.
  - The lasso can create discriminant functions with feature selection.
  - This conceptual framework is referred to as flexible discriminant analysis (FDA).
- If many of the predictors are on different scales, it is difficult for the FDA model to uncover which predictors have the most impact on the response variable (variable importance).

FDA

```
##################################Flexible discriminant analysis##################################
set.seed(476)
FDATune <- train(x = as.matrix(Smarket.train[,1:8]),
y = Smarket.train$Direction,
method = "fda",
preProc = c('center', 'scale'),
metric = "ROC",
trControl = ctrl)
FDATune
```

FDA output

```
>  FDATune
Flexible Discriminant Analysis

998 samples
  8 predictor
  2 classes: 'Down', 'Up'

Pre-processing: centered (8), scaled (8)
Resampling: Repeated Train/Test Splits Estimated (25 reps, 75%)
Summary of sample sizes: 750, 750, 750, 750, 750, 750, ...
Resampling results across tuning parameters:

  nprune  ROC        Sens       Spec
  2       1.0000000  0.9281967  0.9079365
  4       1.0000000  0.9704918  1.0000000
  6       0.9999948  0.9698361  1.0000000

Tuning parameter 'degree' was held constant at a value of 1
ROC was used to select the optimal model using the largest value.
The final values used for the model were degree = 1 and nprune = 2.
```