# Machine Learning for Microeconometrics
# Part 6: Classification and Unsupervised

A. Colin Cameron
Univ.of California - Davis

.

May 2022

# Course Outline

- **1.** Variable selection and cross validation
- **2.** Shrinkage methods
  - ridge, lasso, elastic net
- **3.** ML for causal inference using lasso
  - OLS with many controls, IV with many instruments
- **4.** Other methods for prediction
  - nonparametric regression, principal components, splines
  - neural networks
  - regression trees, random forests, bagging, boosting
- **5:** More ML for causal inference
  - ATE with heterogeneous effects and many controls.
- **Part 6. Classification and unsupervised learning**
  - **classification (categorical $y$) and unsupervised learning (no $y$).**

# 1. Introduction

- To date considered supervised learning with a continuous measure (or a count or binary where model probabilities).
- Now consider very briefly classification and unsupervised learning.
- **Classification** is supervised learning with $y$ categorical
  - The loss function is the number of misclassifications rather than MSE.
  - Traditional methods select the category with the highest predicted probability.
  - Some ML methods instead directly select the category.
- **Unsupervised** learning there is no $y$, only $\mathbf{x}$
  - Principal components.
  - k-means clustering.
- Good reference is ISL2.

# Overview

1. Classification (categorical $y$)
    1. Loss function
    2. Logit
    3. Local logit regression
    4. k-nearest neighbors
    5. Discriminant analysis
    6. Support vector machines
    7. Regression trees and random forests
    8. Neural networks
2. Unsupervised learning (no $y$)
    1. Principal components analysis
    2. Cluster analysis

# 1. Classification: Overview

- **Regression methods**
  - predict probabilities based on log-likelihood rather than MSE
  - assign to class with the highest predicted probability (Bayes classifier)
    - ★ in binary case $\widehat{y} = 1$ if $\widehat{p} \geq 0.5$ and $\widehat{y} = 0$ if $\widehat{p} < 0.5$.
  - parametric: logistic regression, multinomial regression
  - nonparametric: local logit, nearest-neighbors logit
- **Discriminant analysis**
  - additionally assumes a normal distribution for the $\mathbf{x}$'s
  - predict probabilities
  - use Bayes theorem to get $\Pr[Y = k|\mathbf{X} = \mathbf{x}]$ and Bayes classifier.
  - used in many other social sciences

# 1. Classification: Overview (continued)

- **Support vector classifiers** and **support vector machines**
  - directly classify (no probabilities)
  - machine learning methods developed in the 1990's
  - are more nonlinear so may classify better
  - use separating hyperplanes of X and extensions.
- **Random forests**
  - in simplest case minimize the classification error rate rather than the MSE
  - in practice better is to use the Gini index or entropy.
- **Neural networks**
  - can work very well for complex classification such as images.

# 1.1 A Different Loss Function: Error Rate

- Instead of MSE we use the **error rate**
  - the number of misclassifications

$$\text{Error rate} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[y_i \neq \widehat{y}_i],$$

    - ★ where for $K$ categories $y_i = 0, ..., K-1$ and $\widehat{y}_i = 0, ..., K-1$.
    - ★ and indicator $\mathbf{1}[A] = 1$ if event $A$ happens and $= 0$ otherwise.

- The **test error rate** is for the $n_0$ observations in the test sample

$$\text{Ave}(\mathbf{1}[y_0 \neq \widehat{y}_0]) = \frac{1}{n_0}\sum_{i=1}^{n_0} \mathbf{1}[y_{0i} \neq \widehat{y}_{0i}].$$

- Cross validation uses number of misclassified observations. e.g. LOOCV is

$$\text{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} Err_i = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[y_i \neq \widehat{y}_{(-i)}].$$

# Classification Table

- A classification table or confusion matrix is a $K \times K$ table of counts of $(y, \widehat{y})$
- In $2 \times 2$ case with binary $y = 1$ or 0
  - sensitivity is % of $y = 1$ with prediction $\widehat{y} = 1$
  - specificity is % of $y = 0$ with prediction $\widehat{y} = 0$
  - receiver operator characteristics curve (ROC) curve plots sensitivity against 1−sensitivity as threshold for $\widehat{y} = 1$ changes.

# Bayes classifier

- The Bayes classifier selects the most probable class
  - the following gives theoretical justification.
- $L(G, \widehat{G}(\mathbf{x})) = \mathbf{1}[y_i \neq \widehat{y}_i]$
  - $L(G, \widehat{G}(\mathbf{x}))$ is 0 on diagonal of $K \times K$ table and 1 elsewhere
  - where $G$ is actual categories and $\widehat{G}$ is predicted categories.
- Then minimize the expected prediction error

$$\begin{aligned} EPE &= E_{G,\mathbf{x}}[L(G, \widehat{G}(\mathbf{x}))] \\ &= E_{\mathbf{x}}\left[\sum_{k=1}^{K} L(G_k, \widehat{G}(\mathbf{x})) \times \Pr[G_k|\mathbf{x}]\right] \end{aligned}$$

- Minimize EPE pointwise (for each value of $\mathbf{x}$)

$$\begin{aligned} \widehat{G}(\mathbf{x}) &= \arg\min_{g \in G}\left[\sum_{k=1}^{K} L(G_k, g) \times \Pr[G_k|\mathbf{x}]\right] \\ &= \arg\min_{g \in G}[1 - \Pr[g|\mathbf{x}]] \text{ given 0-1 loss} \\ &= \max_{g \in G} \Pr[g|\mathbf{x}] \end{aligned}$$

- So select the most probable class.

# 1.2 Logit

- Directly model $p(\mathbf{x}) = \Pr[y|\mathbf{x}]$.
- Logistic (logit) regression for binary case obtains MLE for

$$\ln\left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})}\right) = \mathbf{x}'\boldsymbol{\beta}.$$

- Statisticians implement using a statistical package for the class of generalized linear models (GLM)
  - logit is in the Bernoulli (or binomial) family with logistic link
  - logit is often the default.
- Logit model is a linear (in $\mathbf{x}$) classifier
  - $\widehat{y} = 1$ if $\widehat{p}(\mathbf{x}) > 0.5$
  - i.e. if $\mathbf{x}'\widehat{\boldsymbol{\beta}} > 0$ since $\widehat{p}(\mathbf{x}) = \Lambda(\mathbf{x}'\widehat{\boldsymbol{\beta}})$ and $\Lambda(0) = \frac{e^0}{1+e^0} = 0.5$.

# Logit Example

- Example considers supplementary health insurance for 65-90 year-olds.

```
. * Data for 65-90 year olds on supplementary insurance indicator and regressors
. use mus203mepsmedexp.dta, clear

. global xlist income educyr age female white hisp marry ///
>    totchr phylim actlim hvgg

. describe suppins $xlist

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------
suppins         float   %9.0g                 =1 if has supp priv insurance
income          double  %12.0g                annual household income/1000
educyr          double  %12.0g                Years of education
age             double  %12.0g                Age
female          double  %12.0g                =1 if female
white           double  %12.0g                =1 if white
hisp            double  %12.0g                =1 if Hispanic
marry           double  %12.0g                =1 if married
totchr          double  %12.0g                # of chronic problems
phylim          double  %12.0g                =1 if has functional limitation
actlim          double  %12.0g                =1 if has activity limitation
hvgg            float   %9.0g                 =1 if health status is excellent,
                                                good or very good
```

# Logit Example (continued)

- Summary statistics

```
. * Summary statistics
. summarize suppins $xlist
```

| Variable | Obs | Mean | Std. Dev. | Min | Max |
|---|---|---|---|---|---|
| suppins | 3,064 | .5812663 | .4934321 | 0 | 1 |
| income | 3,064 | 22.47472 | 22.53491 | -1 | 312.46 |
| educyr | 3,064 | 11.77546 | 3.435878 | 0 | 17 |
| age | 3,064 | 74.17167 | 6.372938 | 65 | 90 |
| female | 3,064 | .5796345 | .4936982 | 0 | 1 |
| white | 3,064 | .9742167 | .1585141 | 0 | 1 |
| hisp | 3,064 | .0848564 | .2787134 | 0 | 1 |
| marry | 3,064 | .5558094 | .4969567 | 0 | 1 |
| totchr | 3,064 | 1.754243 | 1.307197 | 0 | 7 |
| phylim | 3,064 | .4255875 | .4945125 | 0 | 1 |
| actlim | 3,064 | .2836162 | .4508263 | 0 | 1 |
| hvgg | 3,064 | .6054178 | .4888406 | 0 | 1 |

# Logit Example

- Logit model estimates

```
. * logit model
. logit suppins $xlist, nolog

Logistic regression                             Number of obs     =      3,064
                                                LR chi2(11)       =     345.23
                                                Prob > chi2       =     0.0000
Log likelihood = -1910.5353                     Pseudo R2         =     0.0829
```

| suppins | Coef. | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| income | .0180677 | .0025194 | 7.17 | 0.000 | .0131298 | .0230056 |
| educyr | .0776402 | .0131951 | 5.88 | 0.000 | .0517782 | .1035022 |
| age | -.0265837 | .006569 | -4.05 | 0.000 | -.0394586 | -.0137088 |
| female | -.0946782 | .0842343 | -1.12 | 0.261 | -.2597744 | .070418 |
| white | .7438788 | .2441096 | 3.05 | 0.002 | .2654327 | 1.222325 |
| hisp | -.9319462 | .1545418 | -6.03 | 0.000 | -1.234843 | -.6290498 |
| marry | .3739621 | .0859813 | 4.35 | 0.000 | .205442 | .5424823 |
| totchr | .0981018 | .0321459 | 3.05 | 0.002 | .0350971 | .1611065 |
| phylim | .2318278 | .1021466 | 2.27 | 0.023 | .0316242 | .4320315 |
| actlim | -.1836227 | .1102917 | -1.66 | 0.096 | -.3997904 | .0325449 |
| hvgg | .17946 | .0811102 | 2.21 | 0.027 | .0204868 | .3384331 |
| _cons | -.1028233 | .577563 | -0.18 | 0.859 | -1.234826 | 1.029179 |

# Logit Example (continued)

- Classification table manually
  - error rate $= (737 + 347)/3064 = 1084/3064 = 0.354$

```
. * Classification table manually
. predict ph_logit
(option pr assumed; Pr(suppins))

. generate yh_logit = ph_logit >= 0.5

. generate err_logit = (suppins==0 & yh_logit==1) | (suppins==1 & yh_logit==0)

. summarize suppins ph_logit yh_logit err_logit
```

| Variable | Obs | Mean | Std. Dev. | Min | Max |
|---|---|---|---|---|---|
| suppins | 3,064 | .5812663 | .4934321 | 0 | 1 |
| ph_logit | 3,064 | .5812663 | .1609388 | .0900691 | .9954118 |
| yh_logit | 3,064 | .7085509 | .4545041 | 0 | 1 |
| err_logit | 3,064 | .3537859 | .4782218 | 0 | 1 |

```
. tabulate suppins yh_logit
```

| =1 if has supp priv insurance | yh_logit 0 | 1 | Total |
|---|---|---|---|
| 0 | 546 | 737 | 1,283 |
| 1 | 347 | 1,434 | 1,781 |
| Total | 893 | 2,171 | 3,064 |

# Logit Example (continued)

- Classification table using estat classification postestimation command
  - problem: reversed ordering in table makes hard to compare to other models.

```
. * Classification table
. estat classification

Logistic model for suppins

                    True
Classified |         D          ~D |      Total
-----------+----------------------+-----------
         + |      1434         737 |       2171
         - |       347         546 |        893
-----------+----------------------+-----------
     Total |      1781        1283 |       3064

Classified + if predicted Pr(D) >= .5
True D defined as suppins != 0
--------------------------------------------------
Sensitivity                     Pr( +| D)   80.52%
Specificity                     Pr( -|~D)   42.56%
Positive predictive value       Pr( D| +)   66.05%
Negative predictive value       Pr(~D| -)   61.14%
--------------------------------------------------
False + rate for true ~D        Pr( +|~D)   57.44%
False - rate for true D         Pr( -| D)   19.48%
False + rate for classified +   Pr(~D| +)   33.95%
False - rate for classified -   Pr( D| -)   38.86%
--------------------------------------------------
Correctly classified                        64.62%
--------------------------------------------------
```

# 1.3 Nonparametric local logit regression

- Extension of local linear to the logit model.
- At $\mathbf{x} = \mathbf{x}_0$ maximize w.r.t. $\alpha_0$ and $\boldsymbol{\beta}_0$ the weighted logit log density

$$\sum_{i=1}^{n} w_h(\mathbf{x}_i - \mathbf{x}_0) \times \{ y_i \ln \Lambda(\alpha_0 + (\mathbf{x}_i - \mathbf{x}_0)'\boldsymbol{\beta}_0) + (1 - y_i) \ln[1 - \Lambda(\alpha_0 + (\mathbf{x}_i - \mathbf{x}_0)'\boldsymbol{\beta}_0)] \}$$

- Stata add-on command `locreg` in `ivqte` package.

# 1.4 Nonparametric k-nearest neighbors

- For each observation $i$ consider the $K$ neighboring observations that have the closest $\mathbf{x}$ value and estimate $\Pr[Y = j]$ by the fraction of the $K$ neighboring observations with $y = j$.
- k-nearest neighbors (K-NN) for many classes
  - $\Pr[Y = j|\mathbf{x} = \mathbf{x}_0] = \frac{1}{K}\sum_{i \in N_0} \mathbf{1}[y_i = j]$
  - where $N_0$ is the $K$ observations on $\mathbf{x}$ closest to $\mathbf{x}_0$.
- There are many measures of closeness
  - default is Euclidean distance between observations $i$ and $j$

$$\left\{\sum_{a=1}^{p}(x_{ai} - x_{ja})^2\right\}^{1/2} \text{ where there are } p \text{ regressors}$$

- Obtain predicted probabilities
  - then assign to the class with highest predicted probability.

# k-nearest neighbors example

- Here use Euclidean distance and set $K = 11$
  - and results here don't use `looclass` option
  - $584 + 394 = 978$ are misclassified

```
. * K-nn classification table with leave-one out cross validation not as good
. estat classtable, nototals nopercents  // without LOOCV

Resubstitution classification table
```

| Key |
|---|
| Number |

| True suppins | Classified 0 | 1 |
|---|---|---|
| 0 | 889 | 394 |
| 1 | 584 | 1,197 |
| Priors | 0.5000 | 0.5000 |

# 1.5 Linear Discriminant Analysis

- Developed for classification problems such as is a skull Neanderthal or Homo Sapiens given various measures of the skull.
- Discriminant analysis specifies a joint distribution for $(Y, \mathbf{X})$.
- Linear discriminant analysis with $K$ categories
  - assume $\mathbf{X}|Y = k$ is $N(\boldsymbol{\mu}_k, \Sigma)$ with density $f_k(\mathbf{x}) = \Pr[\mathbf{X} = \mathbf{x}|Y = k]$
    - ★ note that only the mean of $\mathbf{X}$ varies with the category $k$
  - and let $\pi_k = \Pr[Y = k]$
- The desired $\Pr[Y = k|\mathbf{X} = \mathbf{x}]$ is obtained using Bayes theorem

$$\Pr[Y = k|\mathbf{X} = \mathbf{x}] = \frac{\Pr[Y = k \ \& \ \mathbf{X} = \mathbf{x}]}{\Pr[\mathbf{X} = \mathbf{x}]} = \frac{\pi_k f_k(\mathbf{x})}{\sum_{j=1}^{K} \pi_j f_j(\mathbf{x})}.$$

- Assign observation $\mathbf{X} = \mathbf{x}$ to class $k$ with largest $\Pr[Y = k|\mathbf{X} = \mathbf{x}]$.

# Linear Discriminant Analysis (continued)

- Upon simplification assignment to class $k$ with largest $\Pr[Y = k|\mathbf{X} = \mathbf{x}]$ is equivalent to choosing model with largest **discriminant function**

$$\delta_k(\mathbf{x}) = \mathbf{x}'\Sigma^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k{}'\Sigma^{-1}\boldsymbol{\mu}_k + \ln \pi_k$$

  - use $\widehat{\boldsymbol{\mu}}_k = \overline{\mathbf{x}}_k$, $\widehat{\Sigma} = \widehat{\text{Var}}[\mathbf{x}_k]$ and $\widehat{\pi}_k = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[y_i = k]$.
- Called linear discriminant analysis as $\delta_k(\mathbf{x})$ linear in $\mathbf{x}$.
  - logit also gives separation linear in $\mathbf{x}$.

# Linear Discriminant Analysis Example

- We have

```
. * Linear discriminant analysis
. discrim lda $xlist, group(suppins) notable

. predict yh_lda
(option classification assumed; group classification)

. estat classtable, nototals nopercents

Resubstitution classification table
```

| Key |
|---|
| Number |

| True suppins | Classified 0 | 1 |
|---|---|---|
| 0 | 770 | 513 |
| 1 | 638 | 1,143 |
| Priors | 0.5000 | 0.5000 |

# Quadratic Discriminant Analysis

- Quadratic discriminant analysis
  - additionally allow different variances so $\mathbf{X}|Y = k$ is $N(\boldsymbol{\mu}_k, \Sigma_k)$
- Upon simplification, the Bayes classifier assigns observation $\mathbf{X} = \mathbf{x}$ to class $k$ which has largest

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\mathbf{x}'\Sigma_k^{-1}\mathbf{x} + \mathbf{x}'\Sigma_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k{}'\Sigma_k^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\ln|\Sigma_k| + \ln \pi_k$$

  - called quadratic discriminant analysis as $\delta_k(\mathbf{x})$ is quadratic in $\mathbf{x}$
- Use rather than LDA only if have a lot of data as requires estimating many parameters.

# Quadratic Discriminant Analysis Example

- We have

```
. * Quadratic discriminant analysis
. discrim qda $xlist, group(suppins) notable

. predict yh_qda
(option classification assumed; group classification)

. estat classtable, nototals nopercents

Resubstitution classification table
```

| Key |
|---|
| Number |

| True suppins | Classified 0 | 1 |
|---|---|---|
| 0 | 468 | 815 |
| 1 | 292 | 1,489 |
| Priors | 0.5000 | 0.5000 |

# LDA versus Logit

- ESL ch.4.4.5 compares linear discriminant analysis and logit
  - Both have log odds ratio linear in $X$
  - LDA is joint model if $Y$ and $X$ versus logit is model of $Y$ conditional on $X$.
  - In the worst case logit ignoring marginal distribution of $X$ has a loss of efficiency of about 30% asymptotically in the error rate.
  - If $X's$ are nonnormal (e.g. categorical) then LDA still doesn't do too bad.

# ISL Figure 4.9: Linear and Quadratic Boundaries

- LDA uses a linear boundary to classify and QDA a quadratic

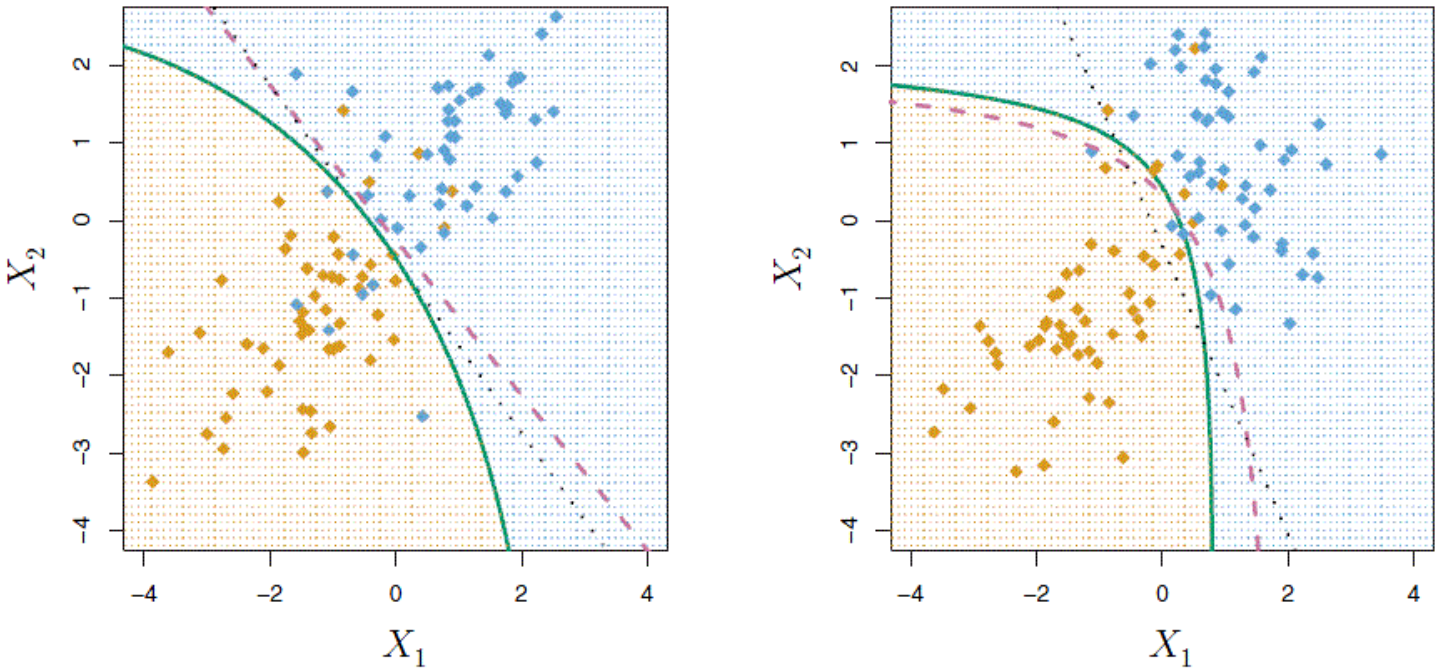


**FIGURE 4.9.** Left: *The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem with* $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$. *The shading indicates the QDA decision rule. Since the Bayes decision boundary is linear, it is more accurately approximated by LDA than by QDA.* Right: *Details are as given in the left-hand panel, except that* $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$. *Since the Bayes decision boundary is non-linear, it is more accurately approximated by QDA than by LDA.*

# 1.6 Support Vector Classifier

- Build on LDA idea of linear boundary to classify when $K = 2$.
- Maximal margin classifier
  - classify using a separating hyperplane (linear combination of $X$)
  - if perfect classification is possible then there are an infinite number of such hyperplanes
  - so use the separating hyperplane that is furthest from the training observations
  - this distance is called the maximal margin.
- Support vector classifier
  - generalize maximal margin classifier to the nonseparable case
  - this adds slack variables to allow some $y$'s to be on the wrong side of the margin
  - $\text{Max}_{\boldsymbol{\beta},\varepsilon} M$ (the margin - distance from separator to training $X$'s) subject to $\boldsymbol{\beta}'\boldsymbol{\beta} \neq \mathbf{1}$, $y_i(\beta_0 + \mathbf{x}_i'\boldsymbol{\beta}) \geq M(1 - \varepsilon_i)$, $\varepsilon_i \geq 0$ and $\sum_{i=1}^{n} \varepsilon_i \leq C$.

# Support Vector Machines

- The support vector classifier has a linear boundary (in $\mathbf{x}_0$)
  - $f(\mathbf{x}_0) = \beta_0 + \sum_{i=1}^{n} \alpha_i \mathbf{x}_0' \mathbf{x}_i$, where $\mathbf{x}_0' \mathbf{x}_i = \sum_{j=1}^{p} x_{0j} x_{ij}$.
- The support vector machine has nonlinear boundaries
  - $f(\mathbf{x}_0) = \beta_0 + \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_0, \mathbf{x}_i)$ where $K(\cdot)$ is a kernel
  - polynomial kernel $K(\mathbf{x}_0, \mathbf{x}_i) = (1 + \sum_{j=1}^{p} x_{0j} x_{ij})^d$
  - radial kernel $K(\mathbf{x}_0, \mathbf{x}_i) = \exp(-\gamma \sum_{j=1}^{p} (x_{0j} - x_{ij})^2)$
- Can extend to $K > 2$ classes (see ISL ch. 9.4).
  - one-versus-one or all-pairs approach
  - one-versus-all approach.

# ISL Figure 9.9: Support Vector Machine

- In this example a linear or quadratic classifier won't work whereas SVM does.

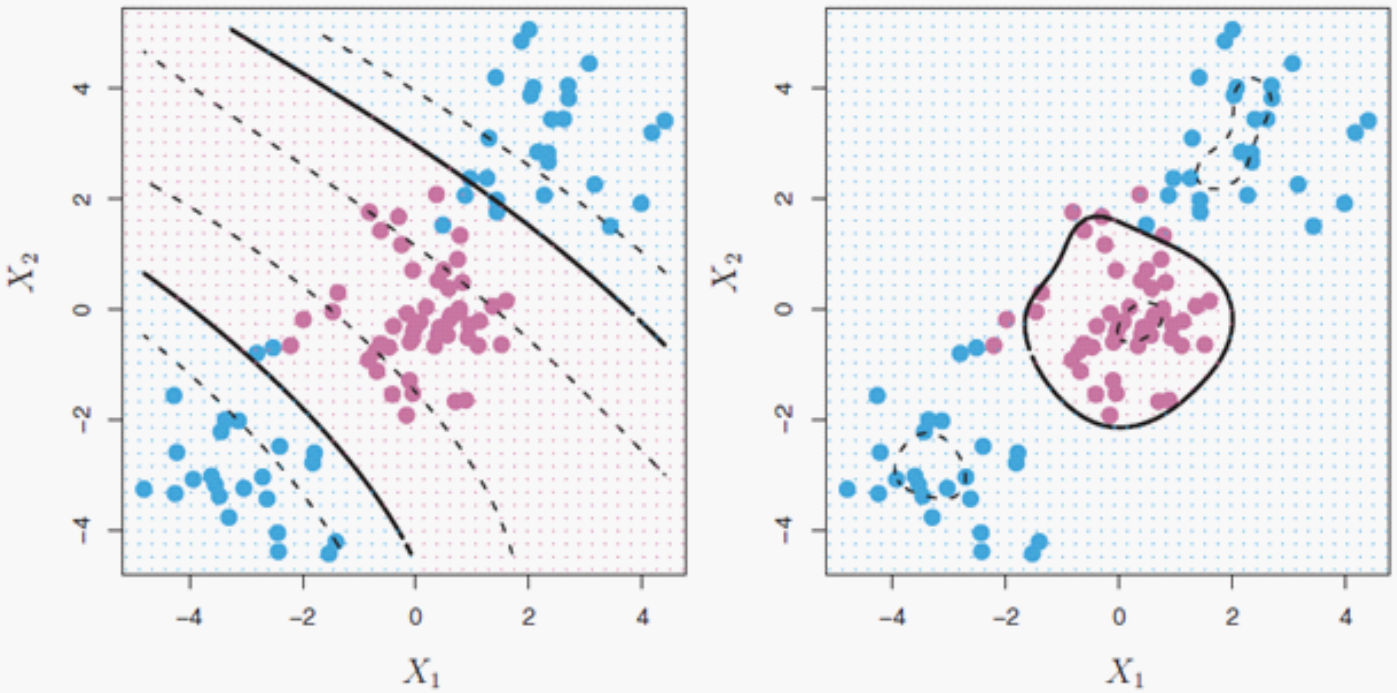


**FIGURE 9.9.** Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

# Support Vector Machines Example

- Use Stata add-on `svmachines` (Guenther and Schonlau)

```
. * Support vector machines - need y to be byte not float and matsize > n
. set matsize 3200

. global xlistshort income educyr age female marry totchr

. generate byte ins = suppins

. svmachines ins income

. svmachines ins $xlist

. predict yh_svm

. tabulate ins yh_svm
```

| ins | yh_svm 0 | 1 | Total |
|---|---|---|---|
| 0 | 820 | 463 | 1,283 |
| 1 | 224 | 1,557 | 1,781 |
| Total | 1,044 | 2,020 | 3,064 |

# Comparison of model predictions

- The following compares the various category predictions.
- SVM does best but we did in-sample predictions here
  - especially for SVM we should have training and test samples.

```
. * Compare various in-sample predictions
. correlate suppins yh_logit yh_knn yh_lda yh_qda yh_svm
(obs=3,064)
```

| | suppins | yh_logit | yh_knn | yh_lda | yh_qda | yh_svm |
|---|---|---|---|---|---|---|
| suppins | 1.0000 | | | | | |
| yh_logit | 0.2505 | 1.0000 | | | | |
| yh_knn | 0.3604 | 0.3575 | 1.0000 | | | |
| yh_lda | 0.2395 | 0.6955 | 0.3776 | 1.0000 | | |
| yh_qda | 0.2294 | 0.6926 | 0.2762 | 0.5850 | 1.0000 | |
| yh_svm | 0.5344 | 0.3966 | 0.6011 | 0.3941 | 0.3206 | 1.0000 |

# 1.7 Regression Trees and Random Forests

- Regression trees, bagging, random forests and boosting can be used for categorical data.
- Let $\widehat{p}_{mk}$ be the proportion of training observations in region $m$ that are from class $k$.
- From ISL2 section 8.1.2 splits can be determined by

$$\begin{array}{ll} \text{Error rate} & 1 - \max_k(\widehat{p}_{mk}) \\ \text{Gini index} & \sum_{k=1}^{K} \widehat{p}_{mk}(1 - \widehat{p}_{mk}) \\ \text{Entropy} & -\sum_{k=1}^{K} \widehat{p}_{mk} \ln \widehat{p}_{mk} \end{array}$$

- Stata user-written `rforest` supports classification in addition to regression.
- Stata user-written `boost` applies to Gaussian (normal), logistic and Poisson regression
  - it uses as loss function for cross-validation the pseudo-$R^2 = 1 - \ln L(\text{full model})/\ln L(\text{intercept-only model})$
  - Matthias Schonlau (2005), The Stata Journal, 5(3), 330-354.

# 1.8 Neural Networks

- Neural networks work very well for classification such as images.

# 2. Unsupervised Learning

- Challenging area: no $y$, only $\mathbf{x}$.
- Example is determining several types of individual based on responses to many psychological questions.
- Principal components analysis.
- Clustering Methods
  - k-means clustering.
  - hierarchical clustering.

# 2.1 Principal Components

- Initially discussed in section on dimension reduction.
- For $p$ regressors goal is to find a few $(m)$ linear combinations of $X$ that explain a good fraction of the total variance $\sum_{j=1}^{p} Var(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$ for mean 0 $X$'s.
- $Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$ where $\sum_{j=1}^{p} \phi_{jm}^2 = 1$ and $\phi_{jm}$ are called factor loadings.
- A useful statistic is the proportion of variance explained (PVE)
  - a scree plot is a plot of $\text{PVE}_m$ against $m$
  - and a plot of the cumulative PVE by $m$ components against $m$.
  - choose $m$ that explains a "sizable" amount of variance
  - ideally find interesting patterns with first few components.
- Easier when used PCA earlier in supervised learning as then observe $Y$ and can treat $m$ as a tuning parameter.
- Stata `pca` command.

# 2.2 Cluster Analysis: k-Means Clustering

- Goal is to find homogeneous subgroups among the $X$.
- K-Means splits into $K$ distinct clusters where within cluster variation is minimized.
- Let $W(C_k)$ be measure of variation
  - Minimize$_{C_1,...,C_k} \sum_{k=1}^{K} W(C_k)$
  - Euclidean distance $W(C_k) = \frac{1}{n_k} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$
- Global maximum requires $K^n$ partitions.
- Instead use algorithm 10.1 (ISL p.388) which finds a local optimum
  - run algorithm multiple times with different seeds
  - choose the optimum with smallest $\sum_{k=1}^{K} W(C_k)$.

# ISL Figure 10.5

- Data is $(x_1.x_2)$ with $K = 2, 3$ and 4 clusters identified.

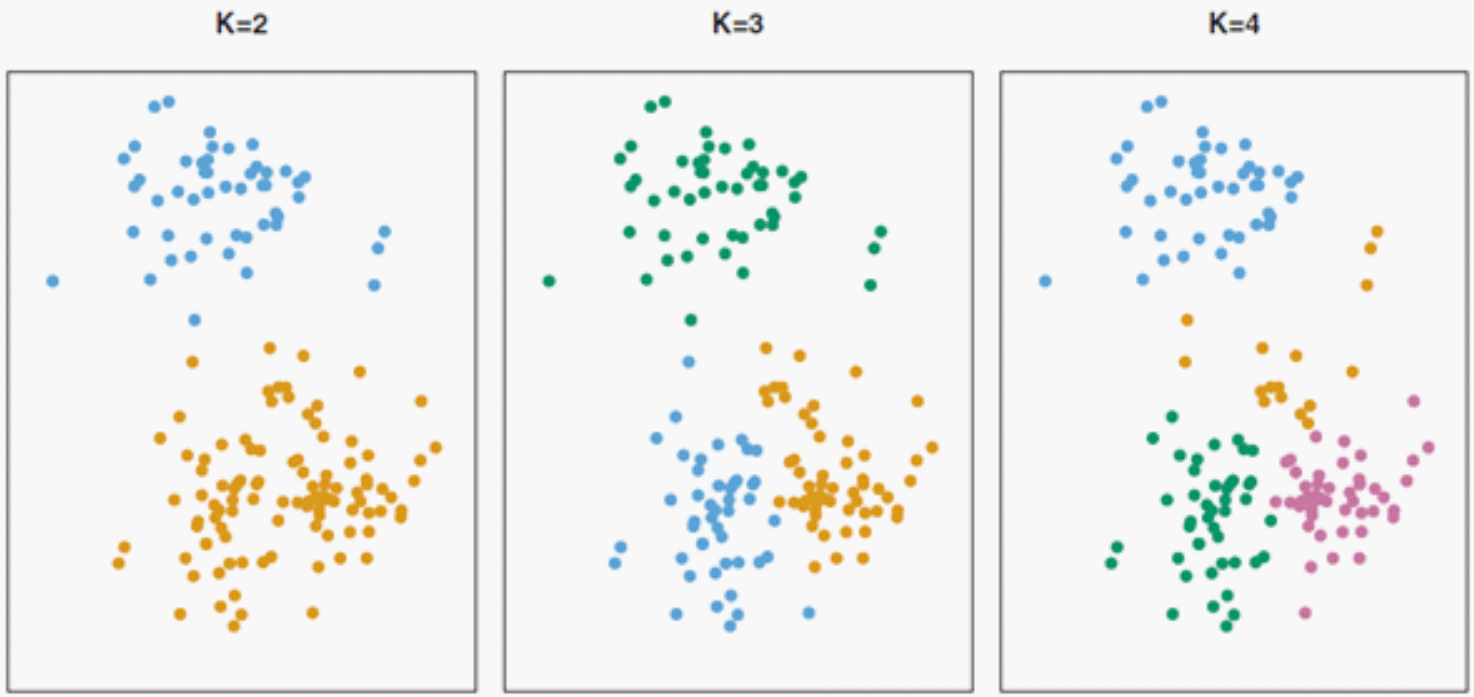


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that*

# k-means Clustering Example

- Use same data as earlier principal components analysis example.

```
. * k-means clustering with defaults and three clusters
. use machlearn_part2_spline.dta, replace

. graph matrix x1 x2 z     // matrix plot of the three variables

. cluster kmeans x1 x2 z, k(3) name(myclusters)

. tabstat x1 x2 z, by(myclusters) stat(mean)

Summary statistics: mean
  by categories of: myclusters
```

| myclusters | x1 | x2 | z |
|---|---|---|---|
| 1 | .8750554 | .503166 | 1.34776 |
| 2 | -.8569585 | -1.120344 | -.5772717 |
| 3 | .1691631 | .6720648 | -.3493614 |
| Total | .0301211 | .0226274 | .0664539 |

# Hierarchical Clustering

- Do not specify $K$.
- Instead begin with $n$ clusters (leaves) and combine clusters into branches up towards trunk
  - represented by a dendrogram
  - eyeball to decide number of clusters.
- Need a dissimilarity measure between clusters
  - four types of linkage: complete, average, single and centroid.
- For any clustering method
  - it is a difficult problem to do unsupervised learning
  - results can change a lot with small changes in method
  - clustering on subsets of the data can provide a sense of robustness.

# 3. Conclusions

- Guard against overfitting
  - use $K$-fold cross validation or penalty measures such as AIC.
- Biased estimators can be better predictors
  - shrinkage towards zero such as Ridge and LASSO.
- For flexible models popular choices are
  - neural nets
  - random forests.
- Though what method is best varies with the application
  - and best are ensemble forecasts that combine different methods.
- Machine learning methods can outperform nonparametric and semiparametric methods
  - so wherever econometricians use nonparametric and semiparametric regression in higher dimensional models it may be useful to use ML methods
  - though the underlying theory still relies on assumptions such as sparsity.

# 4. Software for Machine Learning

- Many ML functions are in Python (`pylearn`) and R.
- Stata 17 covers LASSO, ridge, elastic net, PCA, NP regression, series regression, splines, LDA, QDA, but add-ons are needed for neural networks (`brain`) or random forests (`rforest`) or support vector machines (`svmachines`).
- Stata has integration with Python
  - Giovanni Cerulli (2020), Machine Learning using Stata/Python, https://arxiv.org/pdf/2103.03122v1.pdf
    - ★ Stata add-on r_ml_stata.ado and r_ml_stata.ado are Stata wrappers for tree, boosting, random forest, regularized multinomial, neural network, naive Bayes, nearest neighbor, support vector machine
    - ★ https://sites.google.com/view/giovannicerulli/machine-learning-in-stata
- To run R in Stata the user-written `Rcall` package integrates R within Stata
  - https://github.com/haghish/rcall

# Some R Commands (possibly superseded)

- Basic classification
  - logistic: glm function
  - discriminant analysis: lda() and qda functions in MASS library
  - k nearest neighbors: knn() function in class library
- Support vector machines
  - support vector classifier: svm(... kernel="linear") in e1071 library
  - support vector machine: svm(... kernel="polynomial") or svm(... kernel="radial") in e1071 library
  - receiver operator characteristic curve: rocplot in ROCR library.
- Unsupervised Learning
  - principal components analysis: function prcomp()
  - k-means clustering: function kmeans()
  - hierarchical clustering: function hclust()

# 5. References


- ISL2: Gareth James, Daniela Witten, Trevor Hastie and Robert Tibsharani (2013), An Introduction to Statistical Learning: with Applications in R, second edition, Springer.
  - free legal pdf at https://www.statlearning.com/
- ESL: Trevor Hastie, Robert Tibsharani and Jerome Friedman (2009), The Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer.
  - free legal pdf at http://statweb.stanford.edu/~tibs/ElemStatLearn/index.html
- Chapter 28.6.7-26.8.8 "Machine Learning for prediction and inference" in A. Colin Cameron and Pravin K. Trivedi (2022), Microeconometrics using Stata, Second edition, forthcoming.
  - covers classification and unsupervised learning only very briefly.