Regression and Classification

Big Data Lectures – Chapter 3

Lexin Li

Division of Biostatistics
University of California, Berkeley

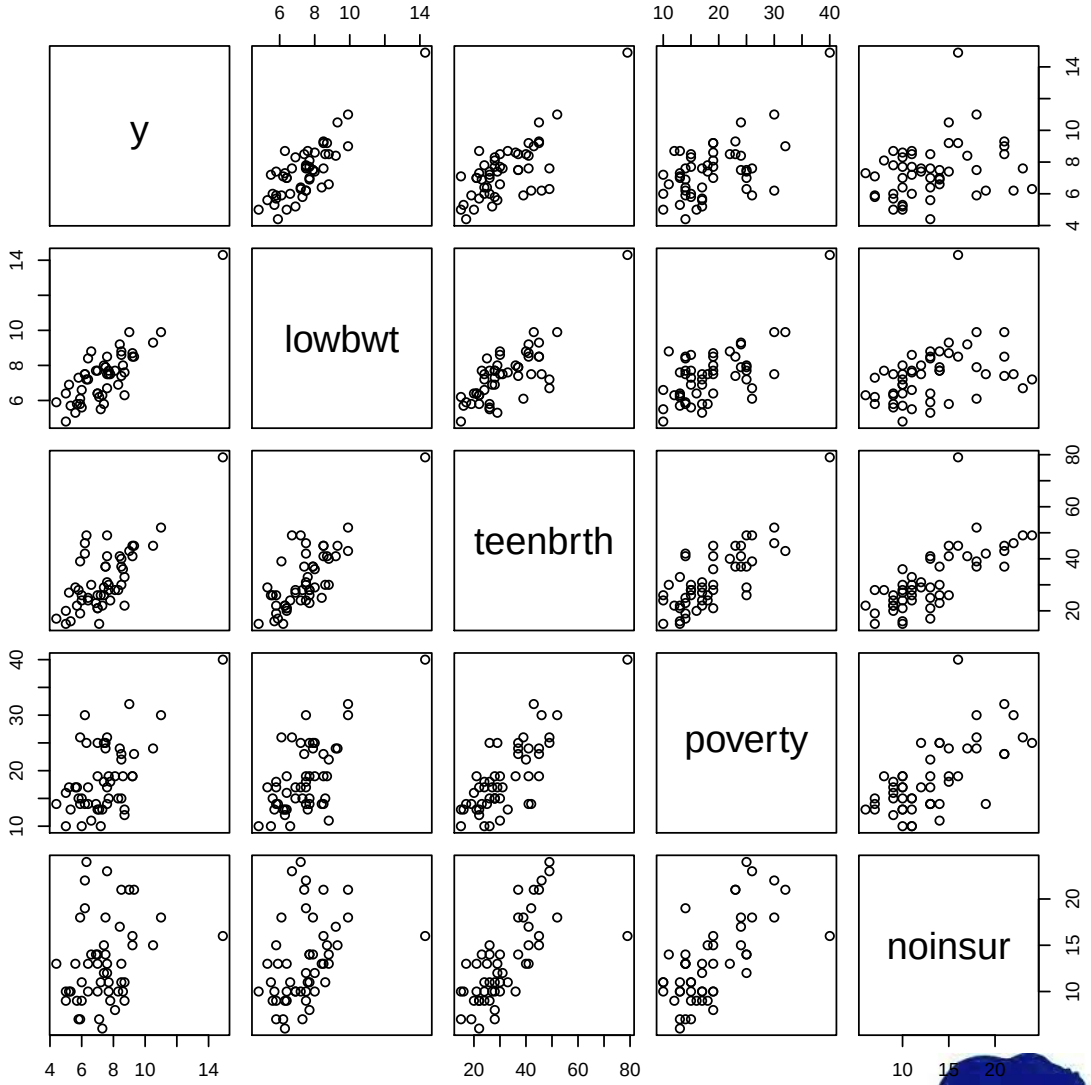
Outline

- list of topics:
  - linear regression
  - logistic regression
  - linear and logistic regression for **Big Data**
    - MapReduce computing
    - p-value: too big to fail
    - sampling: shall we do it?
    - divide-and-conquer
    - heterogeneity: finding subgroups
  - classification: basics
  - discriminant analysis
    - linear / quadratic / mixture discriminant analysis
    - discriminant analysis for **Big Data**
  - nonparametric classification
    - nearest neighbor classifier
    - classification and regression trees
    - linear support vector machines

Linear Regression

Motivating example

- U.S. infant mortality rate from Annie E. Casey Kids Count Data Center
  - Y: infant mortality rate
  - X_1: low birthweight rate
  - X_2: teen birth rate
  - X_3: poverty rate
  - X_4: no insurance rate
  - 50 states + D.C.
  - many other variables



Linear regression

- what is **regression**:
  - response / output / dependent variable $Y \in \mathbb{R}^{r\times 1}$ (focus on $r = 1$)
    predictor / input / feature variable vector $\boldsymbol{X} \in \mathbb{R}^{p\times 1}$ (focus on $p > 1$)
  - regression is about the **association** between Y and $\boldsymbol{X}$; how the value of Y changes as a function of $\boldsymbol{X}$
  - regression is about the **conditional distribution** $Y|\boldsymbol{X}$
  - in most cases, we are interested in $E(Y|\boldsymbol{X})$ (1st moment of $Y|\boldsymbol{X}$); in some cases, we are interested in $\mathrm{var}(Y|\boldsymbol{X})$ (2nd moment)
  - when Y is binary / discrete, regression is also called **classification**

biomarkers that are significantly associated with the outcome

understand the conditional distribution of the outcome given the covariates/predictors - sometimes interested in the 1st moment and sometimes interested in the 2nd moment

Linear regression

- what is **regression**:
  - response / output / dependent variable $Y \in \mathbb{R}^{r\times 1}$ (focus on $r = 1$)
    predictor / input / feature variable vector $\boldsymbol{X} \in \mathbb{R}^{p\times 1}$ (focus on $p > 1$)
  - regression is about the **association** between Y and $\boldsymbol{X}$; how the value of Y changes as a function of $\boldsymbol{X}$
  - regression is about the **conditional distribution** $Y|\boldsymbol{X}$
  - in most cases, we are interested in $E(Y|\boldsymbol{X})$ (1st moment of $Y|\boldsymbol{X}$); in some cases, we are interested in $\mathrm{var}(Y|\boldsymbol{X})$ (2nd moment)
  - when Y is binary / discrete, regression is also called **classification**

- **data visualization**:
  - plot your data, whenever possible!
  - not a trivial task for Big Data!
  - **R:** `ggplot2` **and** `lattice`

Linear regression

- **model**:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

  - population level
  - error ε is assumed $N(0, \sigma^2)$ and is independent of $\boldsymbol{X}$
  - $Y|\boldsymbol{X}$ follows a **normal distribution**
  - $E(Y|\boldsymbol{X}) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ – **linear mean**
  - $\mathrm{var}(Y|\boldsymbol{X}) = \sigma^2$ – **constant variance**

- sample level: given the observed data $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \varepsilon_i, \ i = 1, \ldots, n$$

- matrix form:

$$\mathbf{y}_{n\times 1} = \mathbf{X}_{n\times(p+1)} \boldsymbol{\beta}_{(p+1)\times 1} + \boldsymbol{\varepsilon}_{n\times 1}$$

  - $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$ is the **design matrix**, with 1st column being $\mathbf{1}_{n\times 1}$
  - $\varepsilon \sim N(0, \sigma^2 \boldsymbol{I}_{n\times n})$

Linear regression – interpretation

- **interpretation** of β_j:
  - represents the **partial** effect of X_j on Y, **after** the effect of all other variables have been removed
  - regress the residual $[y_i - \sum_{k=1,k\neq j}^{p} \beta_k x_{ik}]$ on x_{ij} gives the same coefficient β_j

- example:
  - response: infant birthweight (Y)
  - predictors: mother's weight (X_1) + mother's age (X_2) + infant gender (X_3; 1=boy, 0=girl)
  - interpretation of β_1: the average increase of birthweight for one "unit" increase of mother's weight, keeping everything else fixed

Linear regression – estimation

- ordinary least squares:

$$\min_{\beta_0,\beta_1,\ldots,\beta_p} \ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

- matrix form:

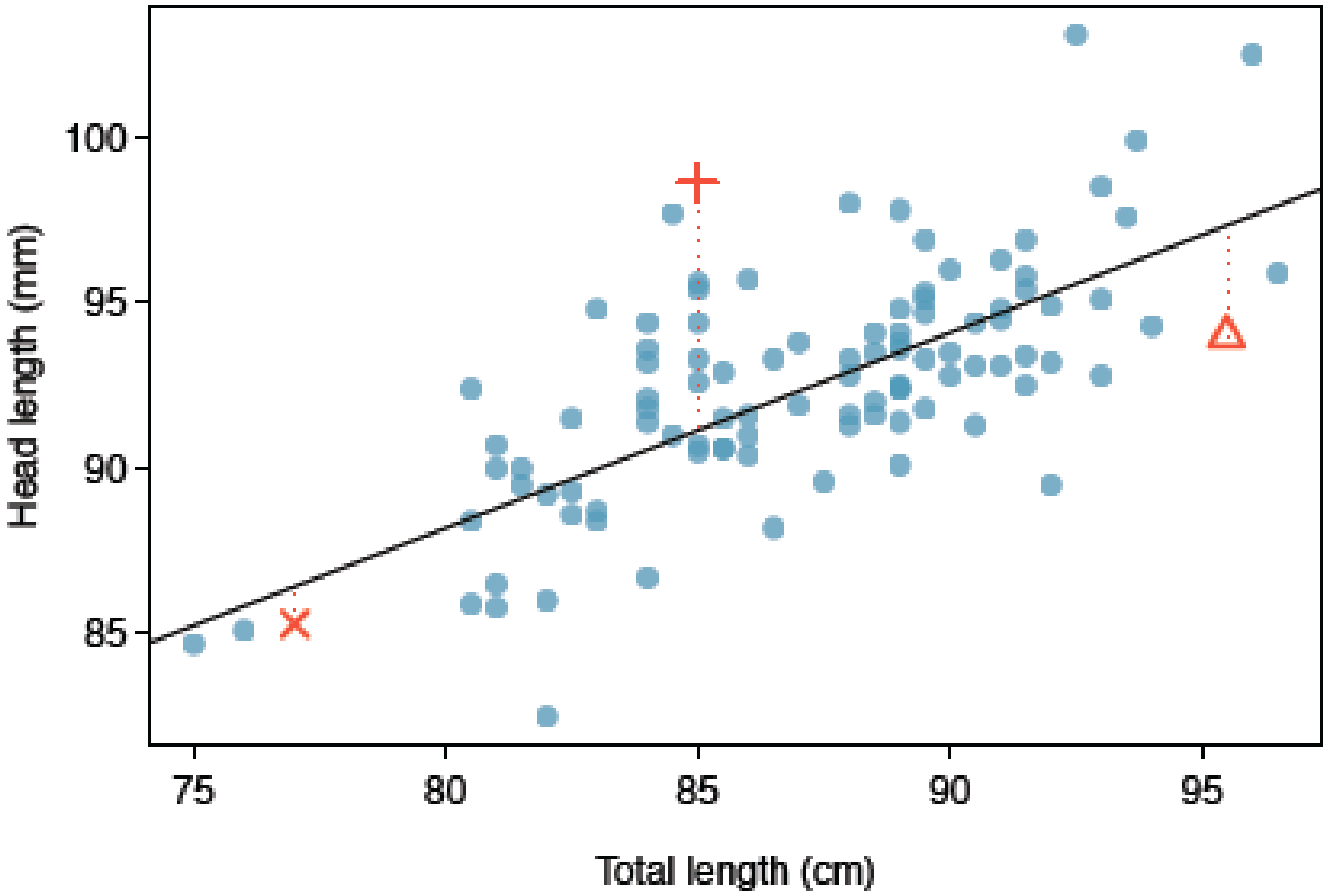
$$\min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathsf{T}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

solution:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y} \\ \hat{\boldsymbol{Y}} &= \mathbf{X}(\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y} = \boldsymbol{H}\mathbf{y} \end{aligned}$$

where $\boldsymbol{H} = \mathbf{X}(\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}$ is the Hat matrix

Linear regression – estimation



Linear regression – estimation

- estimating σ^2:

$$\hat{\sigma}^2 = \frac{SSE}{n-p-1} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-p-1}$$

- goodness of fit: multiple correlation coefficient R^2

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

the proportion of variation in the response that's been accounted for

- adjusted R^2: compensating for more variables

$$R^2 = 1 - \frac{MSE}{MST} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2/(n-p-1)}{\sum_{i=1}^{n}(y_i - \bar{y})^2/(n-1)}$$

- prediction: for an observed sample $\boldsymbol{x}_o = (x_{o1}, \ldots, x_{op})^\mathsf{T}$

$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_{o1} + \ldots \hat{\beta}_p x_{op}$$

Linear regression – estimation

- **maximum likelihood estimation**:

$$\ell(\boldsymbol{\beta}, \sigma^2) = \ln f(Y|\boldsymbol{X}) = \ln \left\{ (2\pi\sigma^2)^{-\frac{n}{2}} \exp \left[\frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\mathsf{T}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{2\sigma^2} \right] \right\}$$

- solution: set $\partial\ell/\partial\boldsymbol{\beta} = 0$ and $\partial\ell/\partial\sigma^2 = 0$

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{MLE} &= (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \hat{\boldsymbol{\beta}}_{OLS} \\ \hat{\sigma}^2_{MLE} &= (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{MLE})^\mathsf{T}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{MLE})/n \neq \hat{\sigma}^2_{OLS} \end{aligned}$$

  which $\hat{\sigma}^2$ is an unbiased estimator?

Linear regression – inference

- **inference, inference, inference**:

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1})$$

- the famous *p*-**value**:

$$\begin{aligned} \text{null hypothesis} \quad & H_0 : \sum_{j=0}^{p} a_j\beta_j = c \\ \text{test statistic} \quad & t = \frac{\sum_{j=0}^{p} a_j\hat{\beta}_j - c}{SE(\sum_{j=0}^{p} a_j\hat{\beta}_j)} \\ p\text{-value} \quad & 2 \times [1 - pt(|t|, n-p-1)] \end{aligned}$$

where $SE(\sum_{j=0}^{p} a_j\hat{\beta}_j) = \{\hat{\sigma}^2\boldsymbol{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\boldsymbol{a}\}^{1/2}$

Linear regression – diagnosis

- question 1: is the linear regression model a good choice?
  - mean function – **linear or nonlinear**
  - variance function – **constant or nonconstant**
  - response distribution – **normal or not**

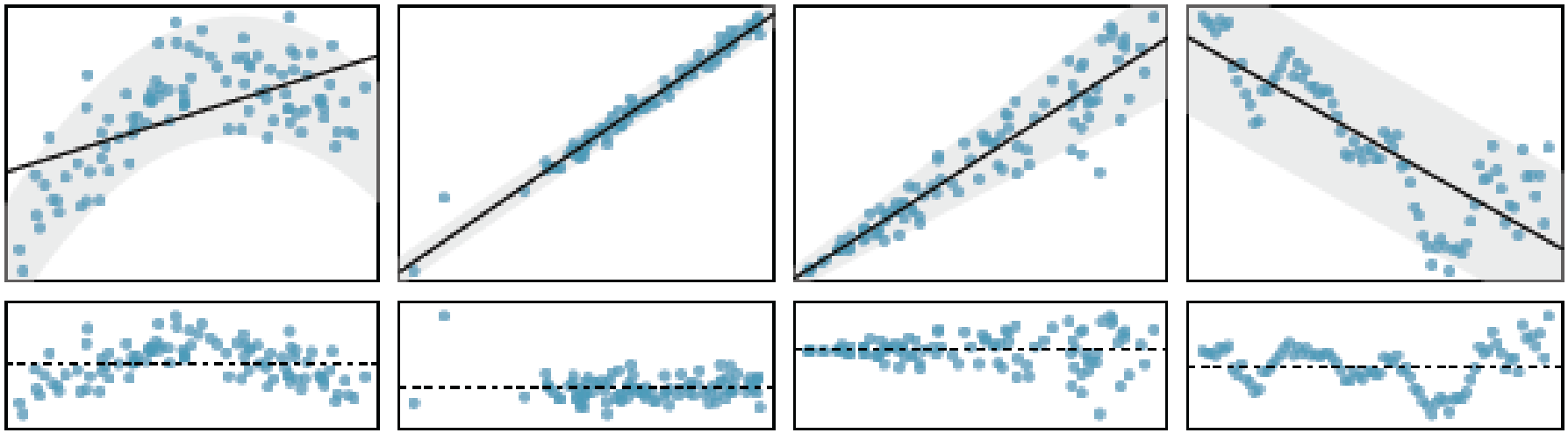
3 assumptions: we care more about the first two (if we have a reasonably large sample size then we don't need to worry too much about the 3rd)

- evaluate the model choice by checking if the assumptions of this model hold (if data don't deviate too far then it's okay)

Linear regression – diagnosis

- question 1: is the linear regression model a good choice?
  - mean function – **linear or nonlinear**
  - variance function – **constant or nonconstant**
  - response distribution – **normal or not**
  - basic idea: If the model is correct, then the **residuals** $e_i = y_i - \hat{y}_i, i = 1, \ldots, n$, should look like a sample from a normal distribution with mean zero and constant variance

residual = observed response - predicted response



residual should capture the behavior of the error (epsilon) if the data is linear

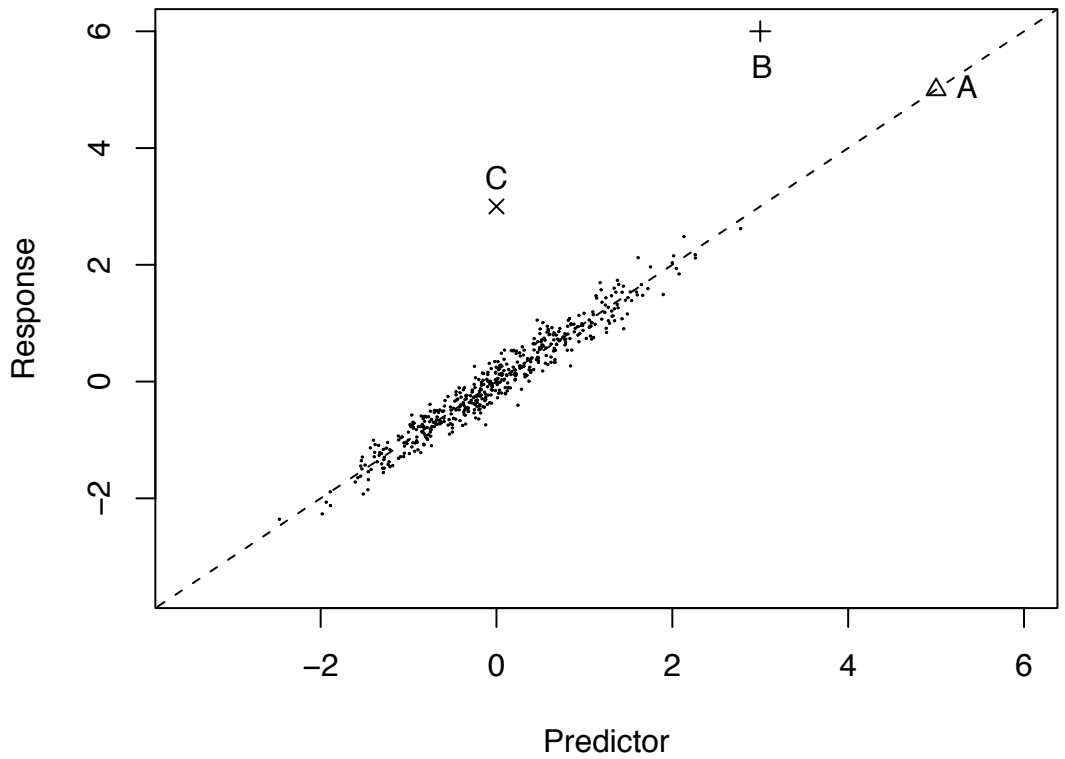
Linear regression – diagnosis

reasonable strategies:
what if you have one random quantity (i.e. IQ) and only 10% of observations have that measurement then you probably shouldn't keep IQ
what if one observation is missing 80% of values? we should probably throw it out

- question 2: is there anything unusual?
  - **influential observation:** data points that influence the regression line the most
  - **outlier:** data points that stand out of the rest – possibly mistakes in data transcription, lab errors, who knows? – those points should be recognized and hopefully explained – they may be what you really want!

DO NOT just throw away outliers!

A, B, and C are outliers
A is not influential but B and C are influential because they would pull the dashed residual line



how do we deal with this assumption checking with big data?
we need to find an alternative way to visualize the data, no good answer

Linear regression – diagnosis

- solution to violation of model assumptions: variable transformation
  - helps achieve linearity; stabilize variance
  - transformation of $\boldsymbol{X}$: $\log(X_j), \sqrt{X_j}, \ldots$
  - transformation of Y: $\log(Y), \sqrt{Y}, \ldots$
- solution to influential observations or outliers:

Linear regression – diagnosis

- solution to violation of model assumptions: variable transformation
  - helps achieve linearity; stabilize variance
  - transformation of $\boldsymbol{X}$: $\log(X_j), \sqrt{X_j}, \ldots$
  - transformation of Y: $\log(Y), \sqrt{Y}, \ldots$
- solution to influential observations or outliers:
  - use intuition, and be careful!

Logistic Regression

Logistic regression

- linear regression model:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

  - $Y|\boldsymbol{X}$ follows a **normal distribution**
  - $E(Y|\boldsymbol{X}) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ – **linear mean**
  - $\text{var}(Y|\boldsymbol{X}) = \sigma^2$ – **constant variance**

— why not use this model for a binary Y?

if Y in binary then YIX follows a BINOMIAL distribution which leads to ALL assumptions for linear regression are violated

We probably need to use logistic regression for binary Y

Logistic regression

- linear regression model:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

  - $Y|\boldsymbol{X}$ follows a **normal distribution**
  - $E(Y|\boldsymbol{X}) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ – **linear mean**
  - $\text{var}(Y|\boldsymbol{X}) = \sigma^2$ – **constant variance**

  — why not use this model for a binary Y?

- **logistic regression model**:

$$\log \frac{P(Y=1|\boldsymbol{X})}{P(Y=0|\boldsymbol{X})} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

  - $Y|\boldsymbol{X}$ follows a **binomial distribution**
  - $E(Y|\boldsymbol{X}) \equiv \mu = P(Y=1|\boldsymbol{X}) = 1/[1+\exp(-\beta^{\mathsf{T}}\boldsymbol{X})]$ – **nonlinear mean**
  - $\text{var}(Y|\boldsymbol{X}) = \mu(1-\mu)$ – **nonconstant variance**

Logistic regression

- logistic regression model:

$$\log \frac{P(Y=1|\boldsymbol{X})}{P(Y=0|\boldsymbol{X})} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

  - **logit** function: $g(\mu) = \log \frac{\mu}{1-\mu}$ = P(Y=1IX) / P(Y=0IX) because Y is binary
  - **link function** connecting $\mu \equiv E(Y|\boldsymbol{X})$ with a **linear combination** of predictors, $\beta^{\mathsf{T}}\boldsymbol{X} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$, i.e.,

$$g(\mu) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

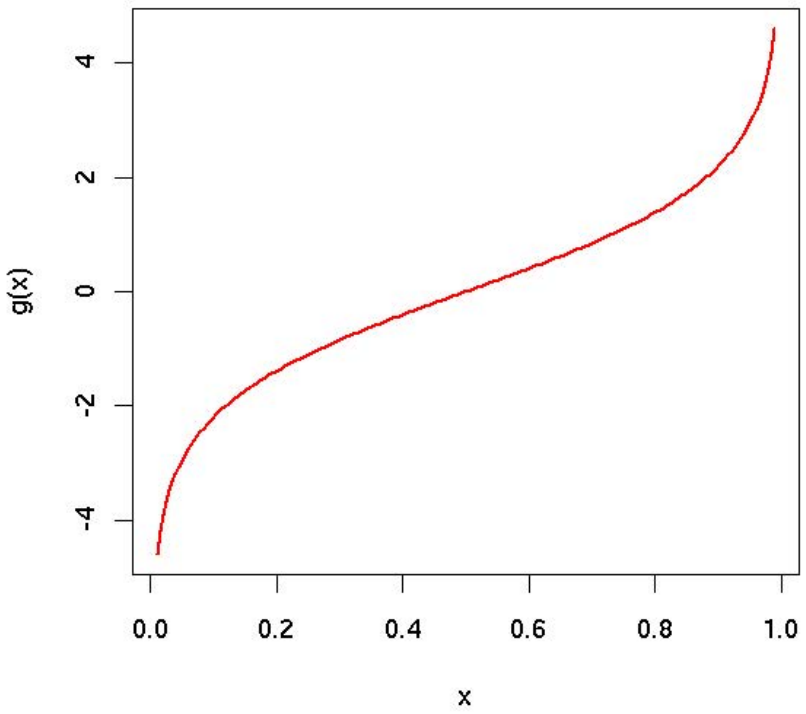
regression is about conditional distribution and we're interested in the E(YIX), can you write explicit formula for the E(YIX)? Yes (b) on next slide

- sample level: given the observed data $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$

$$\log \frac{P(y_i=1|\boldsymbol{x})}{P(y_i=0|\boldsymbol{x})} = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}, \ i = 1, \ldots, n$$

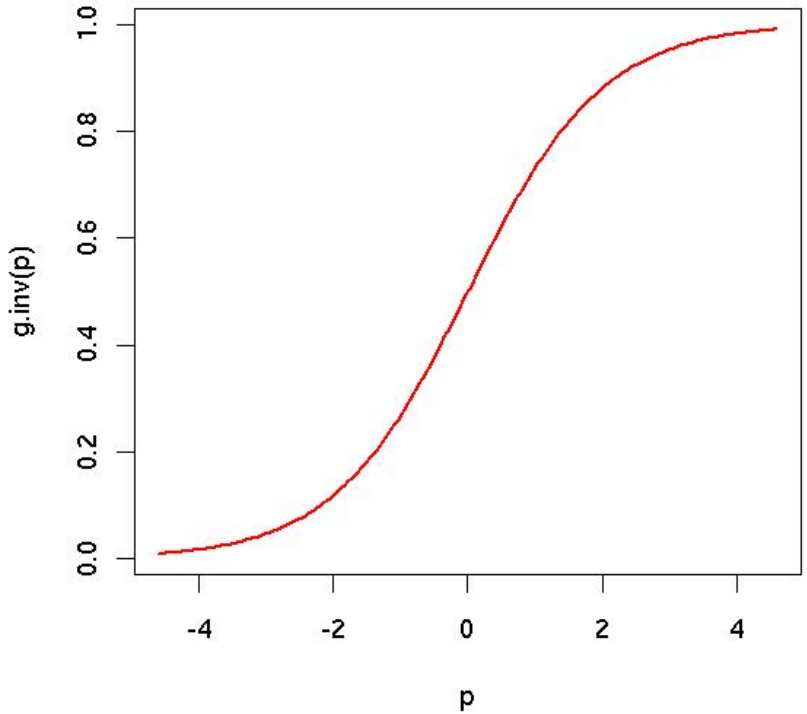
Logistic regression

what if I have count data ? (0,1,2,..) we don't feel comfortable with linear or logistic regression - we fit a Poisson model
how to fit with a linear combination of the predictors?
- we use a generalized linear regression model



(a) logit function $\frac{s}{1-s}$

this graph shows the E(YIX) behavior - the first moment



(b) logistic curve $\frac{1}{1+\exp(-s)}$

x between 0 and 1
outcome unbounded - can take any values on the real line

Logistic regression – interpretation

- **odds**:

$$\text{odds}(A) = \frac{P(A)}{1 - P(A)}$$

- consider a simple example:
  - response: **Disease** $D = 0/1$ (e.g., lung cancer); predictor: **Exposure** $E = 0/1$ (smoking status)
  - logistic regression model:

$$\log \frac{P(D = 1|E)}{P(D = 0|E)} = \beta_0 + \beta_1 E$$

Logistic regression – interpretation

- **odds**:

$$\text{odds}(A) = \frac{P(A)}{1 - P(A)}$$

- consider a simple example:
  - response: **Disease** $D = 0/1$ (e.g., lung cancer); predictor: **Exposure** $E = 0/1$ (smoking status)
  - logistic regression model:

$$\log \frac{P(D = 1|E)}{P(D = 0|E)} = \beta_0 + \beta_1 E$$

  - **odds ratio (OR)**:

$$\begin{aligned} \log(OR) &= \log\left(\frac{\text{odds of } D|E = 1}{\text{odds of } D|E = 0}\right) \\ &= \log(\text{odds of } D|E = 1) - \log(\text{odds of } D|E = 0) \\ &= (\beta_0 + \beta_1 \times 1) - (\beta_0 + \beta_1 \times 0) \\ &= \beta_1 \end{aligned}$$

Logistic regression – interpretation

- interpretation of β_j:
  - the **log Odds Ratio** associated with a unit increase in X_j, with all other predictors fixed
  - for example, in a lung cancer study, $\beta_1 = 5$ means the **odds** of having lung cancer for smokers are $e^5 = 150$ times higher than non-smokers

Logistic regression – interpretation

- interpretation of β_j:
  - the **log Odds Ratio** associated with a unit increase in X_j, with all other predictors fixed
  - for example, in a lung cancer study, $\beta_1 = 5$ means the **odds** of having lung cancer for smokers are $e^5 = 150$ times higher than non-smokers

- a special case: **rare disease hypothesis**
  - when incidence is extremely rare, $P(Y = 0) \approx 1$, then

$$\text{odds} \approx P(Y = 1)$$

  - for example, in a lung cancer study, $\beta_1 = 5$ means the **chance** (probability) of having lung cancer for smokers are $e^5 = 150$ times higher than non-smokers

Logistic regression – estimation and inference

- estimation for linear regression:
  - loss function: $L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$
  - optimization algorithm: ordinary least squares (OLS)
  - closed form solution for $\boldsymbol{\beta}$

Logistic regression – estimation and inference

- estimation for linear regression:
  - loss function: $L(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$
  - optimization algorithm: ordinary least squares (OLS)
  - closed form solution for β

- estimation for logistic regression:

$$L(\beta) = \sum_{i=1}^{n} \left[y_i \log p(\boldsymbol{x}_i; \beta) + (1 - y_i) \log \{1 - p(\boldsymbol{x}_i; \beta)\} \right]$$

where $$p(\boldsymbol{x}_i; \beta) = \mu = \frac{1}{1 + \exp(-\beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})}$$

  - optimization algorithm: iteratively reweighed least squares (IRLS)
  - no closed form solution for β, only the numerical solution

Logistic regression – estimation and inference

- inference for linear regression:

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1})$$

  - test statistic: $t = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$, where $se(\hat{\beta}_j) = \{\hat{\sigma}^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1}_{jj}\}^{1/2}$
  - p-value: $2 \times [1 - pt(|t|, n - p - 1)]$

Logistic regression – estimation and inference

- inference for linear regression:

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1})$$

  - test statistic: $t = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$, where $se(\hat{\beta}_j) = \{\hat{\sigma}^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1}_{jj}\}^{1/2}$
  - p-value: $2 \times [1 - pt(|t|, n - p - 1)]$

- inference for logistic regression:

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, (\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X})^{-1}) \text{ approximately}$$

where $\boldsymbol{W}_{ii} = p(\boldsymbol{x}_i; \hat{\boldsymbol{\beta}})[1 - p(\boldsymbol{x}_i; \hat{\boldsymbol{\beta}})]$

  - test statistic: $t = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$, where $se(\hat{\beta}_j) = \{(\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X})^{-1}_{jj}\}^{1/2}$
  - p-value: $2 \times [1 - pt(|t|, n - p - 1)]$

Logistic regression – estimation and inference

- prediction for linear regression:
  - for an observed sample: $\boldsymbol{x}_o = (x_{o1}, \ldots, x_{op})^\mathsf{T}$

$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_{o1} + \ldots \hat{\beta}_p x_{op}$$

Logistic regression – estimation and inference

- prediction for linear regression:
  - for an observed sample: $\boldsymbol{x}_o = (x_{o1}, \ldots, x_{op})^\mathsf{T}$

$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_{o1} + \ldots \hat{\beta}_p x_{op}$$

- prediction for logistic regression:
  - for an observed sample: $\boldsymbol{x}_o = (x_{o1}, \ldots, x_{op})^\mathsf{T}$

$$\hat{p} = \frac{1}{1 + \exp\{-(\hat{\beta}_0 + \hat{\beta}_1 x_{o1} + \ldots \hat{\beta}_p x_{op})\}}$$

Logistic regression – estimation and inference

- prediction for linear regression:
  - for an observed sample: $\boldsymbol{x}_o = (x_{o1}, \ldots, x_{op})^\mathsf{T}$

$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_{o1} + \ldots \hat{\beta}_p x_{op}$$

- prediction for logistic regression:
  - for an observed sample: $\boldsymbol{x}_o = (x_{o1}, \ldots, x_{op})^\mathsf{T}$

$$\hat{p} = \frac{1}{1 + \exp\{-(\hat{\beta}_0 + \hat{\beta}_1 x_{o1} + \ldots \hat{\beta}_p x_{op})\}}$$

  - choose a threshold value, say, 0.5, then assign $\hat{y} = 1$ if $\hat{p} \geq 0.5$ and $\hat{y} = 0$ if $\hat{p} < 0.5$ $\Rightarrow$ **classification**

Generalized linear regression

- recall: logistic regression model:

$$g(\mu) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

  - **logit** function: $g(\mu) = \log \frac{\mu}{1-\mu}$
  - **link function** connecting $\mu \equiv E(Y|\boldsymbol{X})$ with a **linear combination** of predictors, $\boldsymbol{\beta}^T \boldsymbol{X} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

Generalized linear regression

- recall: logistic regression model:

$$g(\mu) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

  - **logit** function: $g(\mu) = \log \frac{\mu}{1-\mu}$
  - **link function** connecting $\mu \equiv E(Y|\boldsymbol{X})$ with a **linear combination** of predictors, $\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

- **generalized linear model (GLM)**:
  - use a **different link function** connecting $\mu \equiv E(Y|\boldsymbol{X})$ with a **linear combination** of predictors, $\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$
  - include as special cases: linear regression, logistic regression

Generalized linear regression

- recall: logistic regression model:

$$g(\mu) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

  - **logit** function: $g(\mu) = \log \frac{\mu}{1-\mu}$
  - **link function** connecting $\mu \equiv E(Y|\boldsymbol{X})$ with a **linear combination** of predictors, $\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$

- **generalized linear model (GLM)**:
  - use a **different link function** connecting $\mu \equiv E(Y|\boldsymbol{X})$ with a **linear combination** of predictors, $\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X} = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$
  - include as special cases: linear regression, logistic regression

- some commonly used link functions:

link	identity	logit	log	probit	inverse
$g(\mu)$	μ	$\log \frac{\mu}{1-\mu}$	$\log \mu$	$\Phi^{-1}(\mu)$	μ^{-1}

Linear and Logistic Regression for Big Data

MapReduce computing

- linear regression: solve the normal equations $\boldsymbol{A}\beta = \boldsymbol{c}$, where $\boldsymbol{A} = \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}$, $\boldsymbol{c} = \sum_{i=1}^{n} \boldsymbol{x}_i y_i$
- important to note that, $\boldsymbol{A} \in \mathbb{R}^{p \times p}$, n is "big" while p is moderate, so inversion of $\boldsymbol{A}$ is not the bottleneck here
- map-reduce solution:
  - one set of mappers to compute $\sum_{subset} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}$
  - one set of mappers to compute $\sum_{subset} \boldsymbol{x}_i y_i$
  - one reducer to sum up the partial values for $\boldsymbol{A}$
  - one reducer to sum up the partial values for $\boldsymbol{c}$
  - finally compute $\boldsymbol{A}^{-1}\boldsymbol{c}$
- logistic regression: solve an iterative, weighted version

p-value

- example: camera sales on eBay (Lin et al., 2013, Information Systems Research)

$$\begin{aligned}\ln(\text{price}) &= \beta_0 + \beta_1 \ln(\text{minimum bid}) + \beta_2 \text{reserve} \\ &\quad + \beta_3 \ln(\text{seller feedback}) + \beta_4 \text{duration} + \boldsymbol{\gamma}^{\mathsf{T}} \text{controls} + \varepsilon\end{aligned}$$

$n = 341,136$

Variable	Coefficient	Standard error	p-value	95% confidence interval[a]	Interpretation for the conservative bound of the confidence interval for directional hypotheses
ln(minimum bid)	0.1006	0.000825	0.000	(0.0990, 0.1023)	1% increase in the minimum bid is associated with an average 0.09% increase in final price, all else constant.
Reserve	0.7375	0.00675	0.000	(0.7240, 0.7510)	Items with a reserve price sell for a price that is on average 106% ($=100(e^{0.724}-1)\%$) higher, all else constant.
ln(seller feedback)	0.0438	0.00065	0.000	(0.0425, 0.0451)	1% increase in the seller's feedback score is associated with an average of 0.04% higher price, all else constant.
Duration	−0.0405	0.0007	0.000	(−0.0419, −0.0391)	Each extra day for auction listing is associated with an average 4% decrease in price, all else constant.
Control variables: Dummies for camera type, brand, condition, and product lines.					

- what's going on?

p-value

- example: camera sales on eBay (Lin et al., 2013, Information Systems Research)

$$\ln(\text{price}) = \beta_0 + \beta_1 \ln(\text{minimum bid}) + \beta_2 \text{reserve} + \beta_3 \ln(\text{seller feedback}) + \beta_4 \text{duration} + \boldsymbol{\gamma}^{\mathsf{T}} \text{controls} + \varepsilon$$

$n = 341,136$

Variable	Coefficient	Standard error	p-value	95% confidence interval[a]	Interpretation for the conservative bound of the confidence interval for directional hypotheses
ln(minimum bid)	0.1006	0.000825	0.000	(0.0990, 0.1023)	1% increase in the minimum bid is associated with an average 0.09% increase in final price, all else constant.
Reserve	0.7375	0.00675	0.000	(0.7240, 0.7510)	Items with a reserve price sell for a price that is on average 106% ($=100(e^{0.724}-1)\%$) higher, all else constant.
ln(seller feedback)	0.0438	0.00065	0.000	(0.0425, 0.0451)	1% increase in the seller's feedback score is associated with an average of 0.04% higher price, all else constant.
Duration	−0.0405	0.0007	0.000	(−0.0419, −0.0391)	Each extra day for auction listing is associated with an average 4% decrease in price, all else constant.
Control variables: Dummies for camera type, brand, condition, and product lines.					

- what's going on?

$$SE(\hat{\beta}_j) = \{\hat{\sigma}^2 (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}_{jj}\}^{1/2}, \text{ where } \hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-p-1}$$

— too big to fail

p-value

- another example:

```
## fit a linear regression model using bigglm
Large data regression model: bigglm(y ~ x1 + x2 + x3, data = dat)
Sample size = 1.5e+08
               Coef          (95%       CI)           SE         p
(Intercept) -0.0001437 -0.0006601 0.0003727 0.0002582 0.5777919
X1           0.0013703  0.0008047 0.0019360 0.0002828 0.0000013
X2           0.0002371 -0.0003286 0.0008028 0.0002828 0.4018565
X3          -0.0002620 -0.0008277 0.0003037 0.0002829 0.3542728
```

p-value

- another example:

```
## fit a linear regression model using bigglm
Large data regression model: bigglm(y ~ x1 + x2 + x3, data = dat)
Sample size = 1.5e+08
               Coef          (95%       CI)           SE         p
(Intercept) -0.0001437 -0.0006601 0.0003727 0.0002582 0.5777919
X1           0.0013703  0.0008047 0.0019360 0.0002828 0.0000013
X2           0.0002371 -0.0003286 0.0008028 0.0002828 0.4018565
X3          -0.0002620 -0.0008277 0.0003037 0.0002829 0.3542728
```

true model: $Y = 0.001X_1 + \varepsilon$, and $\varepsilon \sim \text{Normal}(0, 1)$

p-value

- another example:

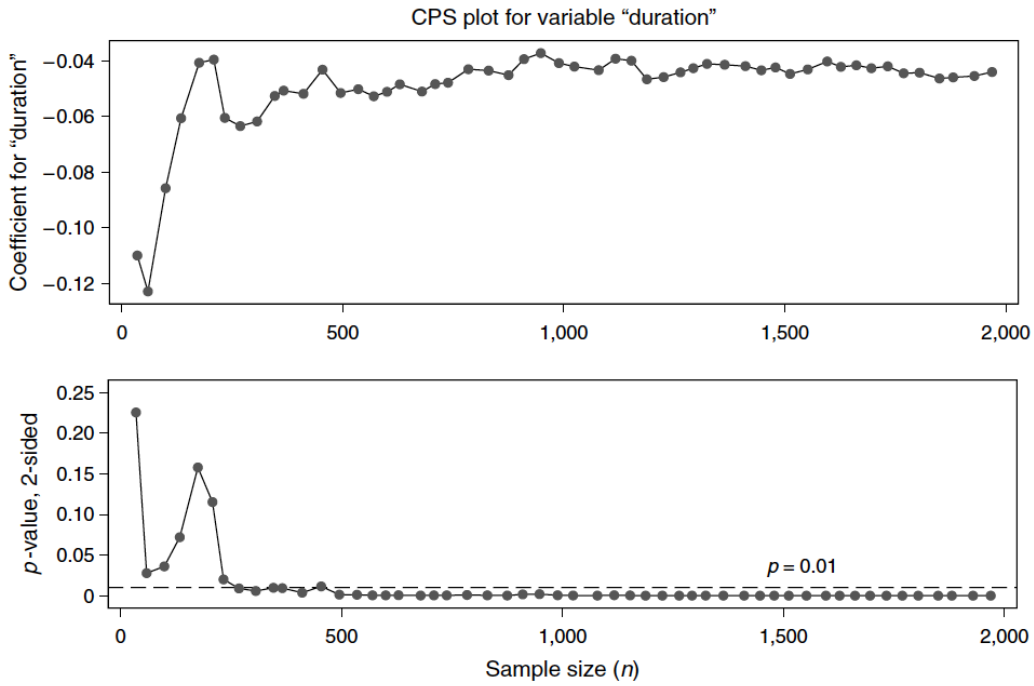
```
## fit a linear regression model using bigglm
Large data regression model: bigglm(y ~ x1 + x2 + x3, data = dat)
Sample size = 1.5e+08
               Coef          (95%       CI)           SE         p
(Intercept) -0.0001437 -0.0006601 0.0003727 0.0002582 0.5777919
X1           0.0013703  0.0008047 0.0019360 0.0002828 0.0000013
X2           0.0002371 -0.0003286 0.0008028 0.0002828 0.4018565
X3          -0.0002620 -0.0008277 0.0003037 0.0002829 0.3542728
```

true model: $Y = 0.001X_1 + \varepsilon$, and $\varepsilon \sim \text{Normal}(0, 1)$

- conclusion:
  - a significant number of papers rely on a low p-value and the sign of a regression coefficient **alone** to support their hypotheses
  - **solely** relying on p-values can lead the researcher to claim support for results of **no practical significance**

p-value

- some potential remedies:
  - be cautious about conclusions based on significance and sign alone
  - report effect size within the study context: change of the dependent variable to changes in the independent variable
  - plot the coefficient of interest, the associated p-value, and the confidence interval as a function of the sample size — **subsampling**
  - does **a single model** really make sense?



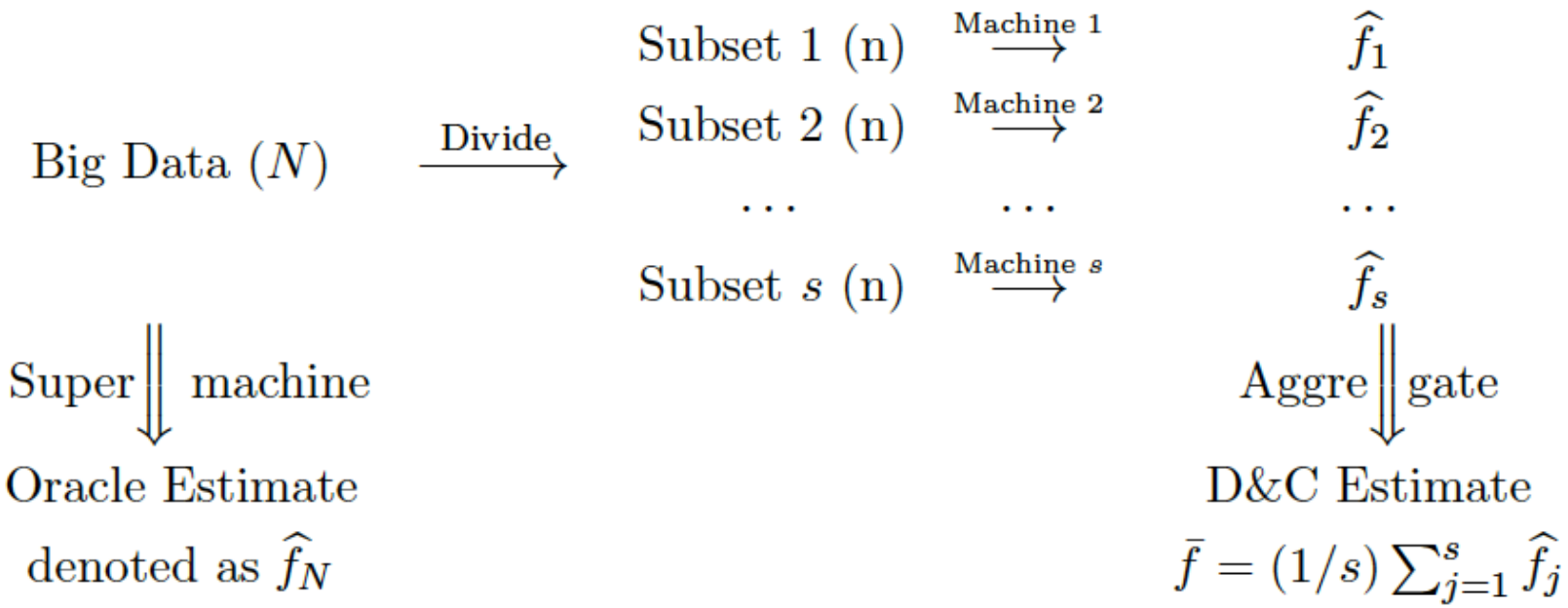
Subsampling

- shall we do it?
  - subsampling for Big Data (Cormode and Duffield, KDD2014)
  - Big Data's small lie – the limitation of sampling and approximation in Big Data analysis (Grey, 2015/07/20)
- how to do it?
  - stream sampling: uniform and weighted case
  - advanced stream sampling: sampling as cost optimization
  - hashing and coordination
  - graph sampling: node, edge and subgraph sampling

Divide-and-conquer

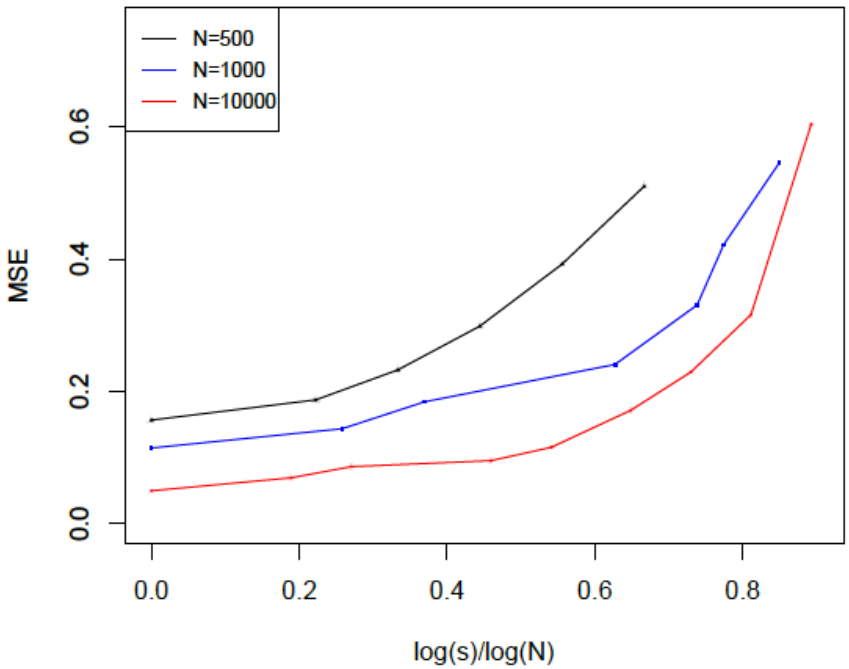
- example: smoothing spline model (Cheng and Shang, 2015, arXiv)

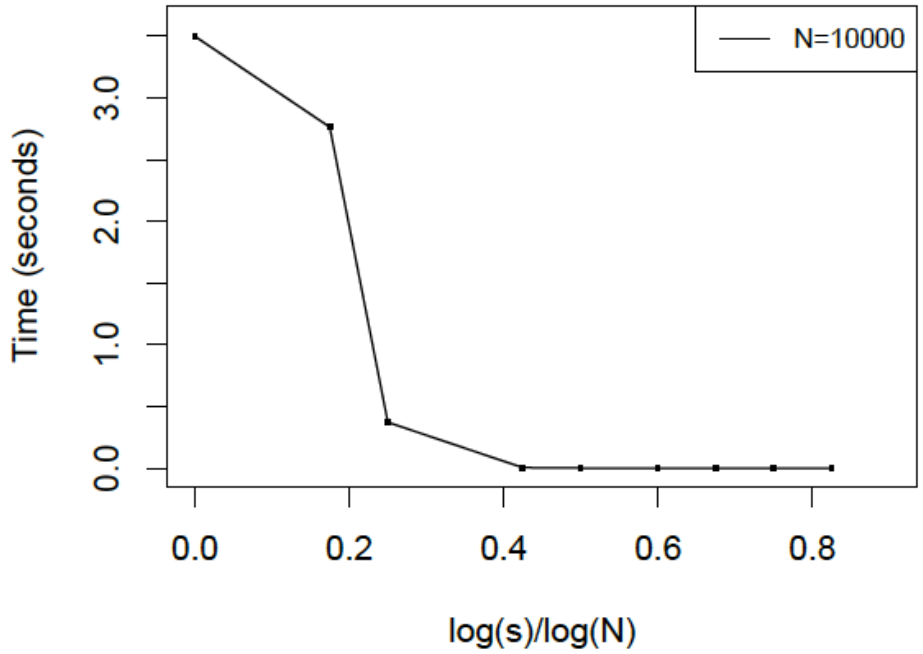
$$Y = f_0(Z) + \varepsilon$$



Divide-and-conquer

- some observations:
  - can reduce computational burden as $\log s / \log N$ increases
  - can preserve statistical efficiency for a wide range of choices of the number of machines s
  - a theoretically interesting question: does there exist a critical value of $\log s / \log N$, beyond which statistical optimality no longer exists?

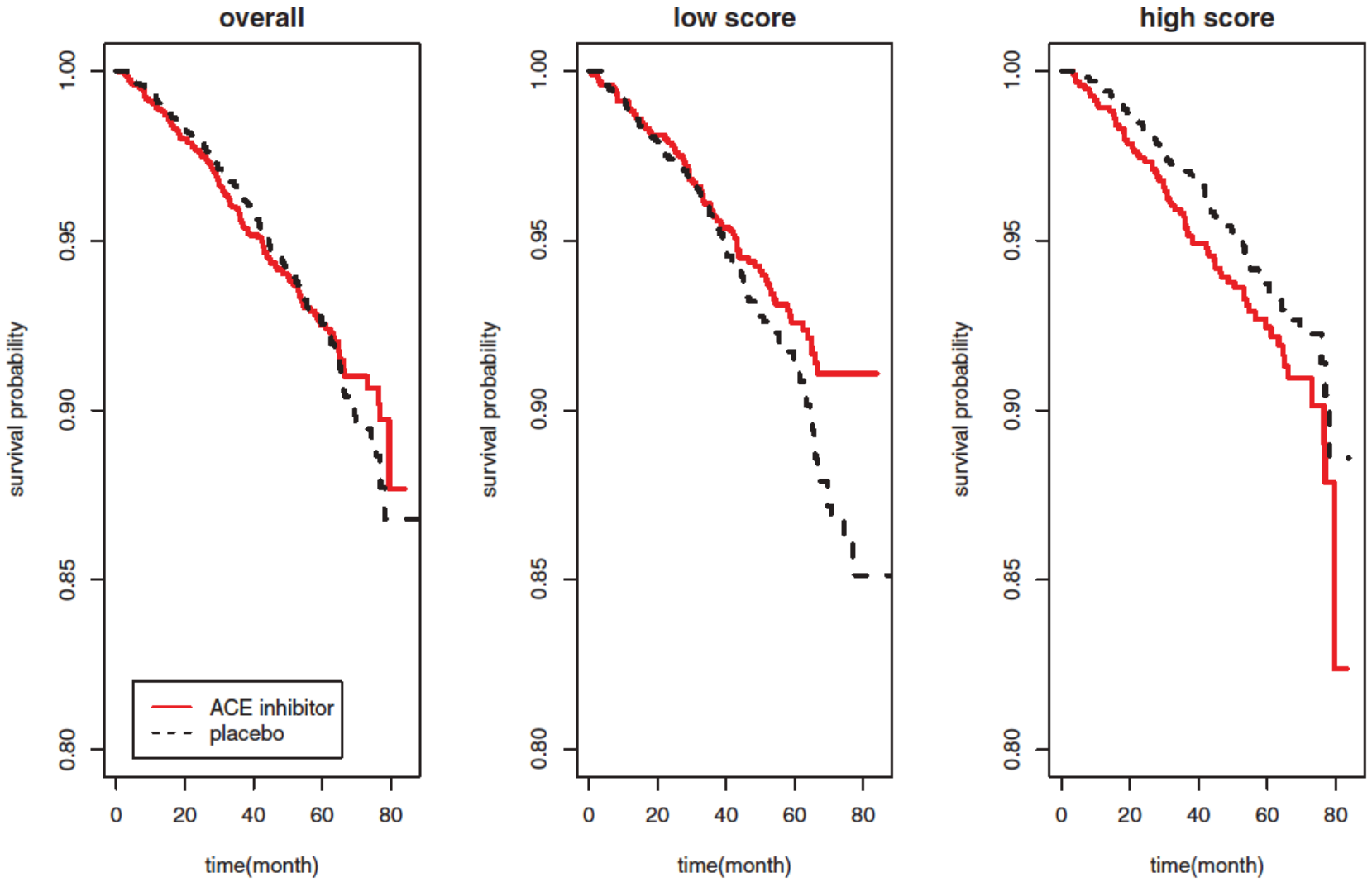




Data heterogeneity

- example: coronary artery disease study (Tian et al., 2014, JASA)
  - patients with stable coronary artery disease and normal or slightly reduced left ventricular function
  - randomly given ACE inhibitor or placebo in a randomized trial
  - constructed a score from baseline covariates on a training set, and then categorized it into low and high
  - plotted the survival curves in a **separate validation** set, **overall and stratified** by the score
  - no significant survival difference between two arms in the overall comparison ($p = 0.67$)
  - patients with low scores have better survival with the ACE inhibitor treatment than with the placebo ($p = 0.06$)
  - this type of information, after appropriate validation, could be very useful in clinical practice
  - in general, **to find the subset of patients** that can potentially benefit from a treatment, determining from a large set of biomarkers

Data heterogeneity



Data heterogeneity

- basic idea:
  - treatment: $T = \pm 1$; potential outcome: $Y^{(1)}, Y^{(-1)}$; baseline covariates: $\boldsymbol{Z}$; functions of covariates: $\boldsymbol{W}(\boldsymbol{Z})$
  - assume the treatment is randomly assigned to a patient, so $T \perp\!\!\!\perp \boldsymbol{Z}$

- **modified outcome model**:

$$\text{linear model:} \qquad Y = \beta_0^{\mathsf{T}} \boldsymbol{W}(\boldsymbol{Z}) + \gamma_0^{\mathsf{T}} \boldsymbol{W}(\boldsymbol{Z}) \cdot T/2 + \varepsilon$$

$$\text{causal treatment effect:} \qquad \Delta(\boldsymbol{z}) = E\left(Y^{(1)} - Y^{(-1)} | \boldsymbol{Z} = \boldsymbol{z}\right) = \gamma_0^{\mathsf{T}} \boldsymbol{W}(\boldsymbol{z})$$

  - the interaction term $\gamma_0^{\mathsf{T}} \boldsymbol{W}(\boldsymbol{z}) \cdot T$ models the heterogeneous treatment effect across the population
  - the linear combination $\gamma_0^{\mathsf{T}} \boldsymbol{W}(\boldsymbol{z})$ can be used for identifying the subgroup of patients who may or may not benefit from the treatment
  - one can estimate γ_0 by minimizing

$$\frac{1}{N} \sum_{i=1}^{N} \left(2 Y_i T_i - \gamma^{\mathsf{T}} \boldsymbol{W}_i\right)^2$$

because $E(2YT|\boldsymbol{Z} = \boldsymbol{z}) = \Delta(\boldsymbol{z})$

Data heterogeneity

- **modified outcome model**:
  - modify the outcome from Y_i to $2Y_iT_i$
  - not easy to extend to other models

- **modified covariate model**:

$$\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \gamma^{\mathsf{T}}\frac{\boldsymbol{W}_i \cdot T_i}{2}\right)^2$$

  - the two estimates are identical and share the same causal interpretations, simply because

$$\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \gamma^{\mathsf{T}}\frac{\boldsymbol{W}_i \cdot T_i}{2}\right)^2 = \frac{1}{4N}\sum_{i=1}^{N}\left(2Y_iT_i - \gamma^{\mathsf{T}}\boldsymbol{W}_i\right)^2$$

  - modify the covariates from $\boldsymbol{W}_i$ to $\boldsymbol{W}_i \cdot T_i/2$
  - easy to extend to binary response, survival response

Classification: Basics

Introduction

- what is **classification**:
  - find the features that best separate a number of **known groups**, and develop a rule to allocate **a new subject** into one of those known groups
  - can be viewed as a regression problem with a categorical response

- examples:
  - click / no click of an online ads; buy / no buy of a product; fraud / no fraud of a credit card transaction; bankrupt / no bankrupt of a business; survival / no survival of a cancer in 5 years; types of cancer
  - a majority of machine learning problems take the form of classification; extremely important in a large variety of applications
  - Cleveland heart disease data:
    – response: presence (1) / absence (0) of heart disease
    – predictors: $p = 13$, age, gender, chest pain type, blood pressure, ...
    – $n = 303$ patients

Evaluation

- misclassification error rate

Evaluation

- misclassification error rate
  - example 1: truth 50 $+1$ and 50 -1; classifier 55 $+1$ and 45 -1

Evaluation

- misclassification error rate
  - example 1: truth 50 $+1$ and 50 -1; classifier 55 $+1$ and 45 -1
  - example 2: truth 95 $+1$ and 5 -1; classifier 100 $+1$ and 0 -1

Evaluation

- misclassification error rate
  - example 1: truth 50 $+1$ and 50 -1; classifier 55 $+1$ and 45 -1
  - example 2: truth 95 $+1$ and 5 -1; classifier 100 $+1$ and 0 -1

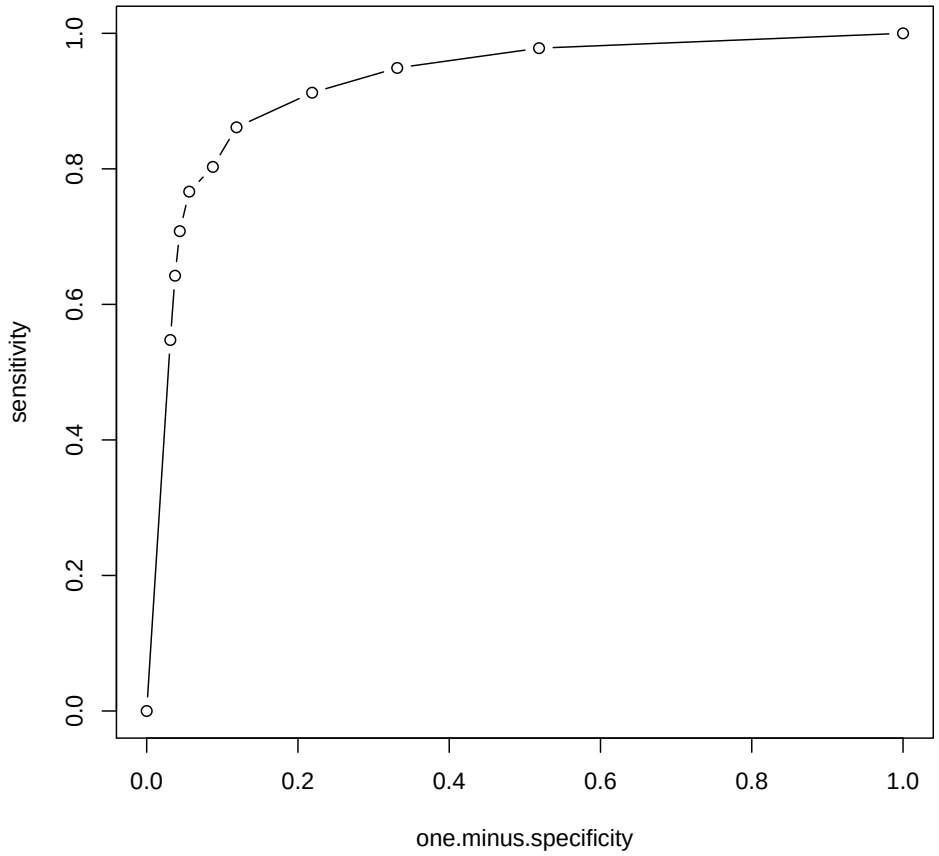
- a 2×2 table:

		Classification		
		1	0	
Truth	1	TP	FN	P
	0	FP	TN	N

  - **sensitivity** = TP / P = probability of predicting 1 given truth is 1
    1 - specificity = FP / N = probability of predicting 1 given truth is 0
  - precision = TP / (TP + FP)
    recall = TP / P (= sensitivity)
  - false positive rate = FP / N (= 1 - specificity)

Evaluation

- receiver operating characteristic (ROC) curve
  - plot of sensitivity versus (1 - specificity)
  - AUC: area under the ROC curve



Evaluation

- example: Cleveland heart disease data

```
cut.off<-seq(0,1,by=0.1)
sensitivity<-NULL
one.minus.specificity<-NULL
for(i in 1:length(cut.off)) {
    Y.hat<-rep(0,length(Y))
    Y.hat[fitted(fit.glm2) >= cut.off[i]]<-1
    sensitivity<-c(sensitivity, sum((Y==1)&(Y.hat==1))/sum(Y==1))
    one.minus.specificity<-c(one.minus.specificity,
                  sum((Y==0)&(Y.hat==1))/sum(Y==0))
}
plot(one.minus.specificity, sensitivity, type="b")
```

Evaluation

- example: Cleveland heart disease data

```
cut.off<-seq(0,1,by=0.1)
sensitivity<-NULL
one.minus.specificity<-NULL
for(i in 1:length(cut.off)) {
    Y.hat<-rep(0,length(Y))
    Y.hat[fitted(fit.glm2) >= cut.off[i]]<-1
    sensitivity<-c(sensitivity, sum((Y==1)&(Y.hat==1))/sum(Y==1))
    one.minus.specificity<-c(one.minus.specificity,
                  sum((Y==0)&(Y.hat==1))/sum(Y==0))
}
plot(one.minus.specificity, sensitivity, type="b")
```

- more about evaluation:
  - **cost** of misclassification: credit card fraud detection – classify a fraud as a normal transaction *vs* classify a normal transaction as a fraud
  - **extremely unbalanced** data: credit card fraud detection; rare disease with incidence rate 1 out of 1,000
  - **training** misclassification error and **testing** misclassification error

Two-class classification

- problem formulation:
  - response: $Y = 0/1$ or $Y = \pm 1$
  - predictor vector: $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^{\mathsf{T}}$
  - goal: **classify Y given $\boldsymbol{X}$**
  - prior class probability: $p_1 = P(Y = 1)$ and $p_0 = P(Y = 0)$
  - conditional density of $\boldsymbol{X}|Y$: $f_1(\boldsymbol{x}) = f(\boldsymbol{x}|Y = 1)$ and $f_0(\boldsymbol{x}) = f(\boldsymbol{x}|Y = 0)$

Two-class classification

- problem formulation:
  - response: $Y = 0/1$ or $Y = \pm 1$
  - predictor vector: $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^{\mathsf{T}}$
  - goal: **classify Y given $\boldsymbol{X}$**
  - prior class probability: $p_1 = P(Y = 1)$ and $p_0 = P(Y = 0)$
  - conditional density of $\boldsymbol{X}|Y$: $f_1(\boldsymbol{x}) = f(\boldsymbol{x}|Y = 1)$ and $f_0(\boldsymbol{x}) = f(\boldsymbol{x}|Y = 0)$
- Bayes classification rule:
  - classify to class 1 if $P(Y = 1|\boldsymbol{X} = \boldsymbol{x}) \geq P(Y = 0|\boldsymbol{X} = \boldsymbol{x})$ because of intuition; e.g., logistic regression

Two-class classification

- problem formulation:
  - response: $Y = 0/1$ or $Y = \pm 1$
  - predictor vector: $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^{\mathsf{T}}$
  - goal: **classify Y given $\boldsymbol{X}$**
  - prior class probability: $p_1 = P(Y = 1)$ and $p_0 = P(Y = 0)$
  - conditional density of $\boldsymbol{X}|Y$: $f_1(\boldsymbol{x}) = f(\boldsymbol{x}|Y = 1)$ and $f_0(\boldsymbol{x}) = f(\boldsymbol{x}|Y = 0)$

- Bayes classification rule:
  - classify to class 1 if $P(Y = 1|\boldsymbol{X} = \boldsymbol{x}) \geq P(Y = 0|\boldsymbol{X} = \boldsymbol{x})$ because of intuition; e.g., logistic regression
  - classify to class 1 if $f_1(\boldsymbol{x}) > f_0(\boldsymbol{x})$ (assuming $p_1 = p_0$) because of

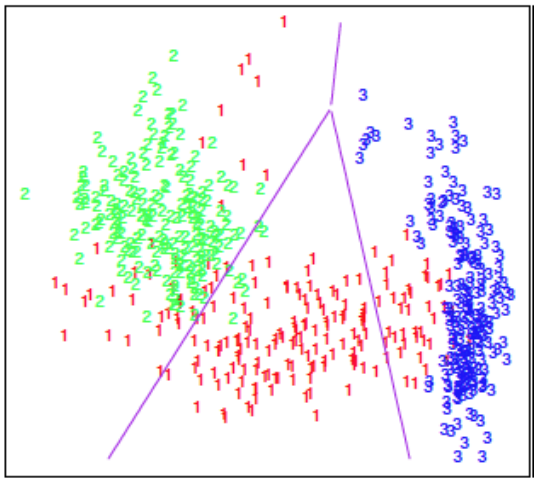
$$P(Y|\boldsymbol{X}) = \frac{P(X|Y)\, P(Y)}{P(\boldsymbol{X})}$$

Discriminant Analysis

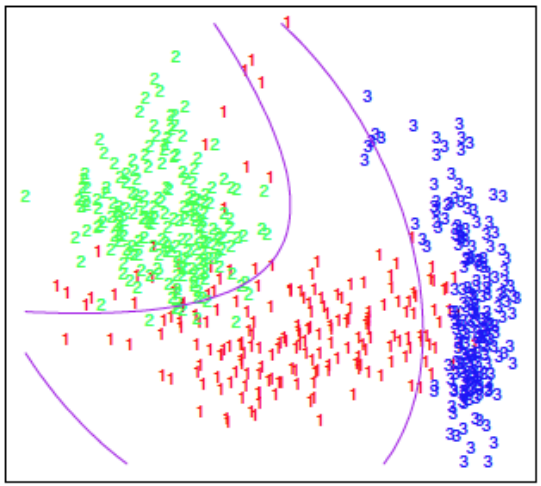
Discriminant analysis

- discriminant analysis:
  - **key idea**: impose a model on $P(\boldsymbol{X}|Y)$, then evaluate $f_1(\boldsymbol{x})/f_0(\boldsymbol{x})$
  - linear / quadratic / mixture discriminant analysis (LDA / QDA / MDA)
  - logistic regression vs LDA

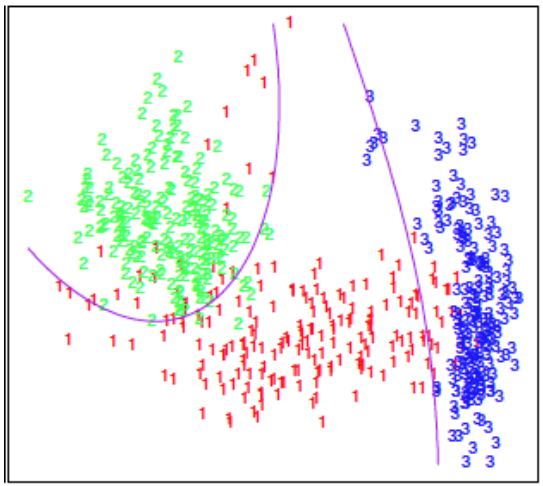
LDA on x_1, x_2



LDA on x_1, x_2
x_1x_2, x_1^2, x_2^2



QDA on x_1, x_2



Linear discriminant analysis

- LDA:
  - key assumption: **normal distribution with equal variance**

$$\begin{aligned} \boldsymbol{X}|Y=1 &\sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \\ \boldsymbol{X}|Y=0 &\sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}) \end{aligned}$$

  - log ratio of the class densities:

$$\ln\left[\frac{f_1(\boldsymbol{x})}{f_0(\boldsymbol{x})}\right] = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)\right]$$

  - Bayes rule:

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)\right] \geq \ln \frac{p_0}{p_1}$$

  - when $p_0 = p_1$:

$$(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \geq (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)$$

    let $\boldsymbol{a} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1}$ — all it counts is $\boldsymbol{a}^{\mathsf{T}}\boldsymbol{x}$

$$\mathbf{a}^{\mathsf{T}}\mathbf{x} \geq \frac{1}{2}(\mathbf{a}^{\mathsf{T}}\boldsymbol{\mu}_1 + \mathbf{a}^{\mathsf{T}}\boldsymbol{\mu}_0)$$

Quadratic discriminant analysis

- QDA:
  - key assumption: : **normal distribution with unequal variance**

$$\begin{aligned} \boldsymbol{X}|Y=1 &\sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \boldsymbol{X}|Y=0 &\sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \end{aligned}$$

  - log ratio of the class densities:

$$\ln\left[\frac{f_1(\boldsymbol{x})}{f_0(\boldsymbol{x})}\right] = \text{ a quadratic function of } \boldsymbol{x}$$

  - Bayes rule:

$$-\frac{1}{2}\boldsymbol{x}^{\mathsf{T}}(\boldsymbol{\Sigma}_1^{-1}-\boldsymbol{\Sigma}_0^{-1})\boldsymbol{x}+\left(\boldsymbol{\mu}_1'\boldsymbol{\Sigma}_1^{-1}-\boldsymbol{\mu}_0\boldsymbol{\Sigma}_0^{-1}\right)^{\mathsf{T}}\boldsymbol{x}-c \geq \ln\frac{p_0}{p_1}$$

where $c = \frac{1}{2}\ln\left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\right)+\frac{1}{2}\left(\boldsymbol{\mu}_1^{\mathsf{T}}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1-\boldsymbol{\mu}_0^{\mathsf{T}}\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right)$

Mixture discriminant analysis

- linear discriminant analysis:

$$\begin{aligned} \boldsymbol{X}|Y=1 &\sim \phi(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \\ \boldsymbol{X}|Y=0 &\sim \phi(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}) \end{aligned}$$

- quadratic discriminant analysis:

$$\begin{aligned} \boldsymbol{X}|Y=1 &\sim \phi(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \boldsymbol{X}|Y=0 &\sim \phi(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \end{aligned}$$

- **mixture discriminant analysis:**

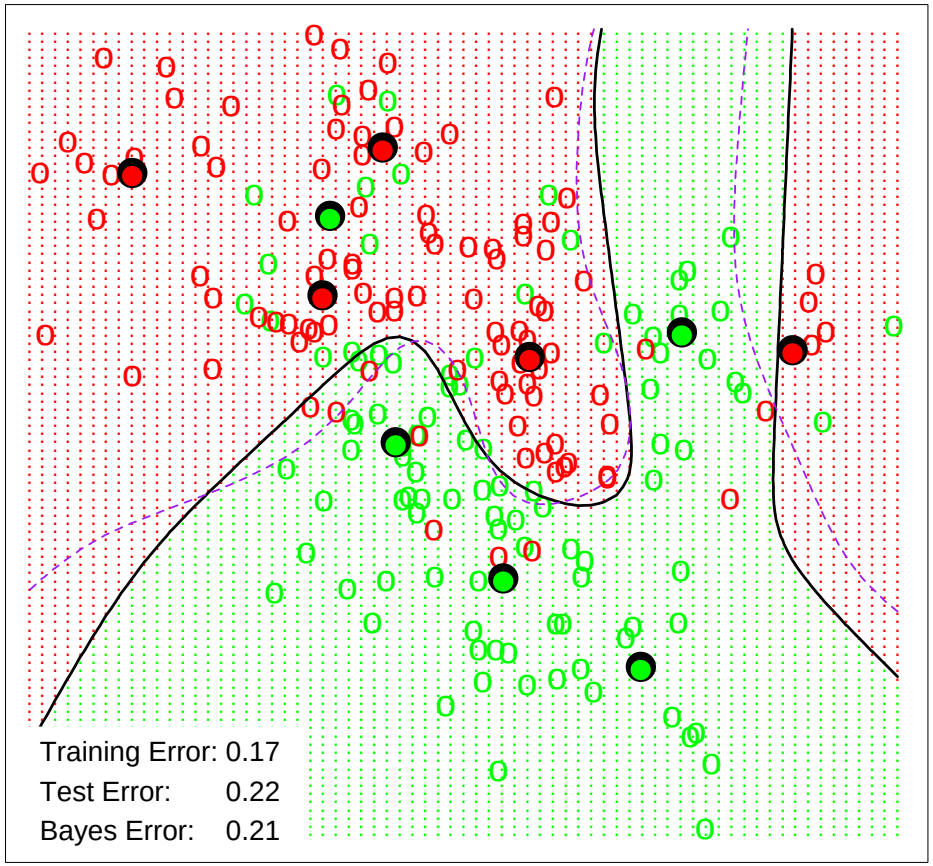
$$\begin{aligned} \boldsymbol{X}|Y=1 &\sim \sum_{r=1}^{R_1} \pi_{1r} \phi(\boldsymbol{\mu}_{1r}, \boldsymbol{\Sigma}) \\ \boldsymbol{X}|Y=0 &\sim \sum_{r=1}^{R_0} \pi_{0r} \phi(\boldsymbol{\mu}_{0r}, \boldsymbol{\Sigma}) \end{aligned}$$

Mixture discriminant analysis

- remarks:
  - LDA: a **single** normal density
    MDA: a **mixture** of normal densities
  - LDA: a **linear** decision boundary
    MDA: a **nonlinear** decision boundary
  - a **common covariance** matrix $\mathbf{\Sigma}$ is still assumed in MDA
  - MDA includes LDA as a special case, when $R_1 = R_0 = 1$, and is generally more flexible than LDA
  - estimation is achieved through the **EM algorithm**
  - limitations: heavily depends on the starting values, often unstable

- generalizations to **more than 2 classes**:
  - LDA, QDA, MDA: all straightforward; pairwise comparison

Mixture discriminant analysis



Discriminant analysis for Big Data

- big n:
  - consider LDA as an example, key components include:

$$p_1 = \sum_{i=1}^n I(y_i = 1) \qquad p_0 = \sum_{i=1}^n I(y_i = 0)$$
$$\boldsymbol{\mu}_1 = \sum_{i=1}^n I(y_i = 1)\boldsymbol{x}_i \qquad \boldsymbol{\mu}_0 = \sum_{i=1}^n I(y_i = 0)\boldsymbol{x}_i$$
$$\boldsymbol{\Sigma} = \sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}$$

  - map-reduce solution:
    - separate sets of mappers to compute the partial sums for $p_1, p_0, \boldsymbol{\mu}_1, \boldsymbol{\mu}_0$, and $\boldsymbol{\Sigma}$, respectively
    - separate reducers to sum up the partial values for $p_1, p_0, \boldsymbol{\mu}_1, \boldsymbol{\mu}_0$, and $\boldsymbol{\Sigma}$, respectively
    - form the LDA decision rule
  - ideas for QDA, MDA are similar
- big p:
  - sparse regularization
  - Mai and Zou (2013)

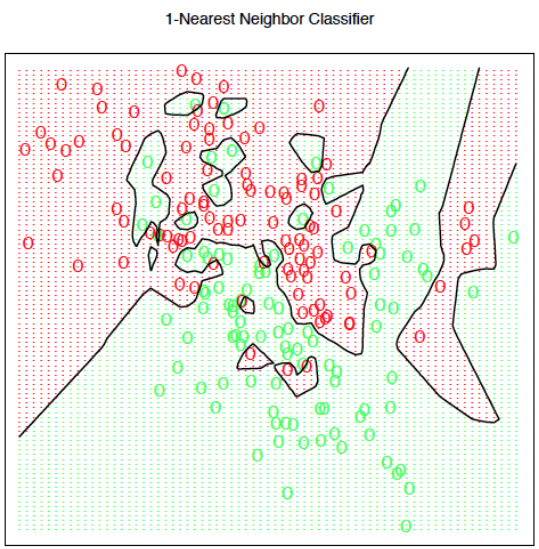
Nonparametric Classification

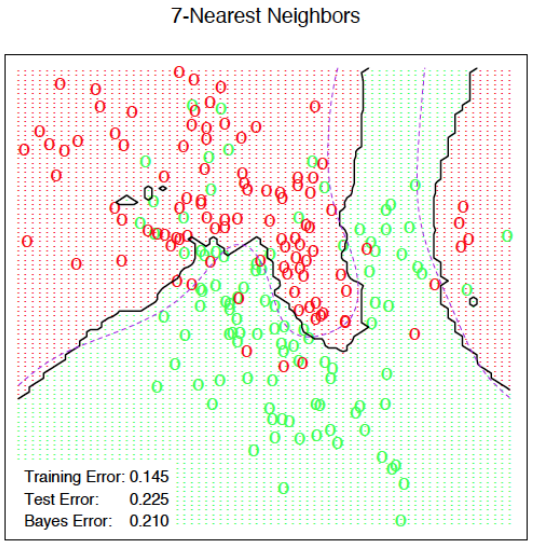
Nearest neighbor classifier

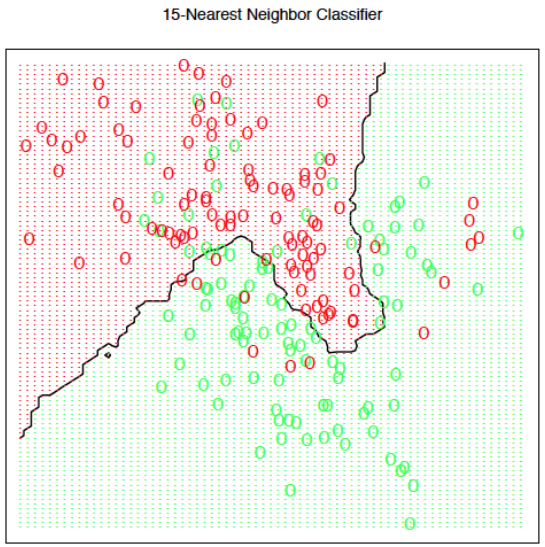
- **nearest neighbor classifier:**
  - given a new observation $\boldsymbol{x}_0$, find k training observations $\boldsymbol{x}_{(i)}$, $i = 1, \ldots, k$, that are **closest** to $\boldsymbol{x}_0$, and then classify using the **majority vote** among these k **neighbors**.

- remarks:
  - key ingredient: a **distance measure**, e.g., the Euclidean distance $\{(\boldsymbol{x}_{(i)} - \boldsymbol{x})^{\mathsf{T}}(\boldsymbol{x}_{(i)} - \boldsymbol{x})\}^{1/2}$.
  - Ican also be applied to non-real-valued $\boldsymbol{x}$ as long as a distance measure can be defined.
  - no probability model is assumed – completely **nonparametric**
  - this seemingly simple classifier works amazingly well in a lot of real applications

Nearest neighbor classifier



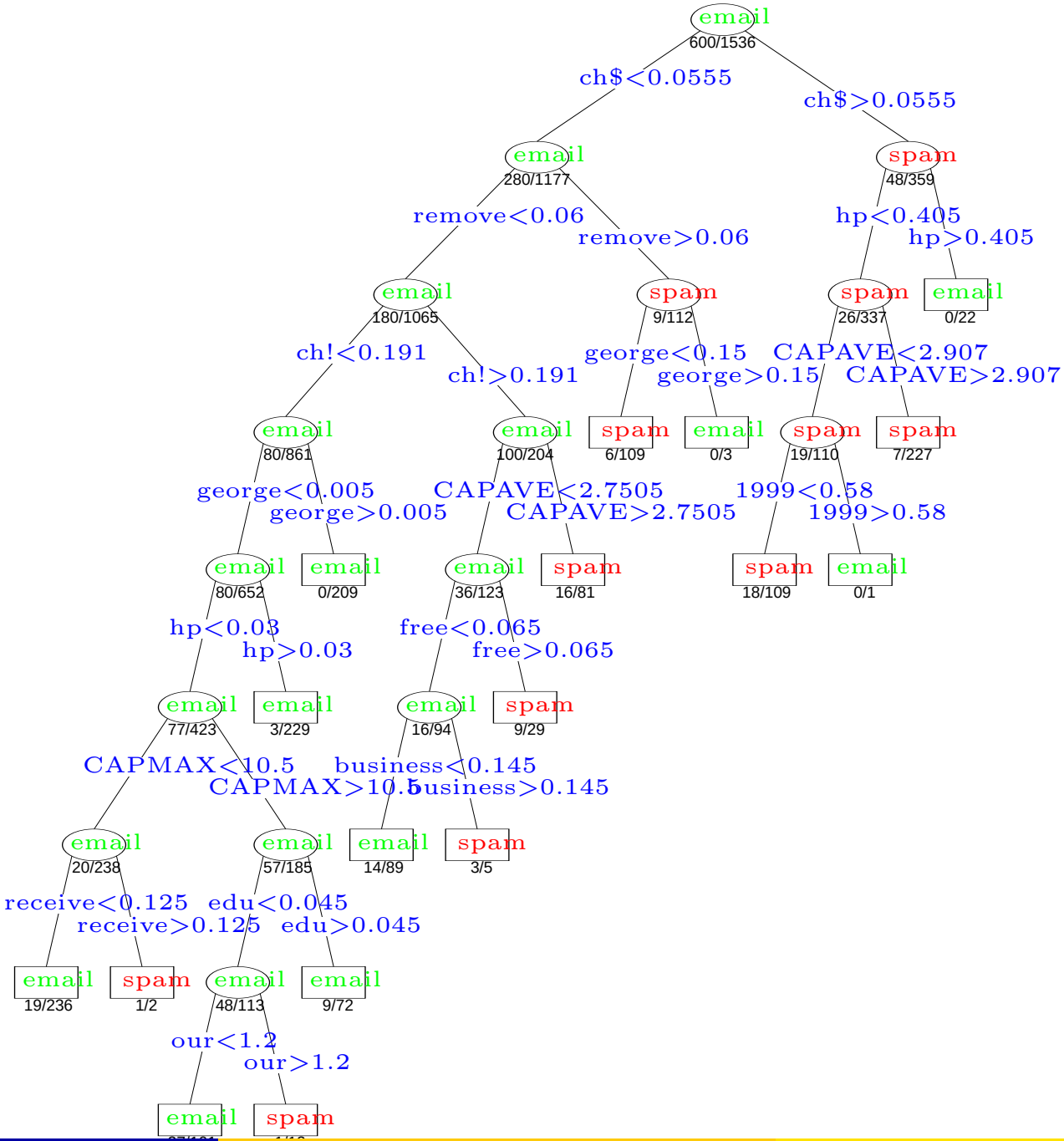




Classification and regression trees

- classification tree:
  - initially all objects are considered as a single group.
  - the group is **binary split** into two subgroups using $X_j \geq c$ for one group and $X_j < c$ for the other group for a selected X_j.
  - the splitting process stops until some stopping criterion is met.

- main steps:
  - split process: choose splitting variables and split points
  - partition process: partition the data into two regions and repeat the splitting process on each region
  - pruning process: a very large tree might overfit the data, while a small tree might not capture the important structure – grow a large tree then prune the tree


email
600/1536
ch$<0.0555
ch$>0.0555
email
280/1177
spam
48/359
remove<0.06
remove>0.06
hp<0.405
hp>0.405
email
180/1065
spam
9/112
spam
26/337
email
0/22
ch!<0.191
ch!>0.191
george<0.15
george>0.15
CAPAVE<2.907
CAPAVE>2.907
email
80/861
email
100/204
spam
6/109
email
0/3
spam
19/110
spam
7/227
george<0.005
george>0.005
CAPAVE<2.7505
CAPAVE>2.7505
1999<0.58
1999>0.58
email
80/652
email
0/209
email
36/123
spam
16/81
spam
18/109
email
0/1
hp<0.03
hp>0.03
free<0.065
free>0.065
email
77/423
email
3/229
email
16/94
spam
9/29
CAPMAX<10.5
CAPMAX>10.5
business<0.145
business>0.145
email
20/238
email
57/185
email
14/89
spam
3/5
receive<0.125
receive>0.125
edu<0.045
edu>0.045
email
19/236
spam
1/2
email
48/113
email
9/72
our<1.2
our>1.2
email
spam

Classification and regression trees

- advantages of CART:
  - very easy to interpret, especially thanks to the binary split
  - handle missing values in a natural way – create a "missing" category
  - handle categorical and ordered variables in a simple and natural way
  - automatic stepwise variable selection and complexity reduction

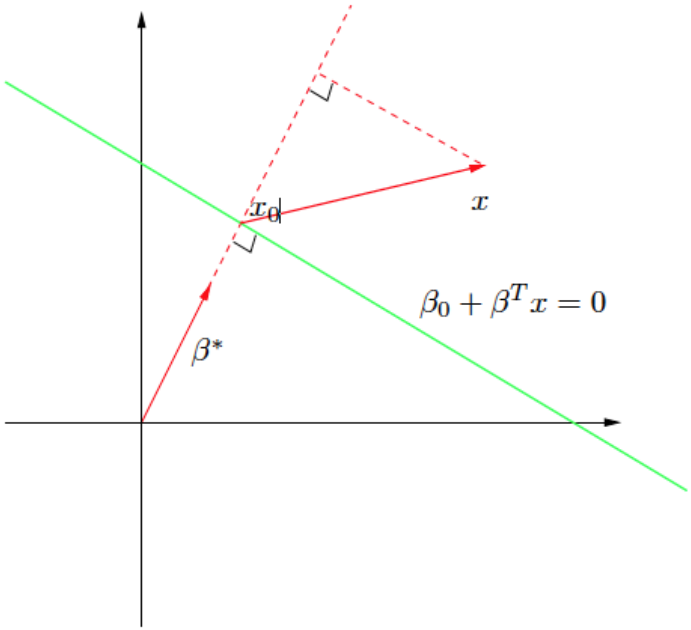
- disadvantages of CART:
  - high variance:
    - a small change in the data can sometimes result in a very different series of splits, making interpretations somewhat difficult
    - main reason is the hierarchical nature of the process: the effect of an error in the top split is propagated down to all the splits below it
    - **bagging**: averages many trees to reduce the variance
  - low accuracy:
    - lack of smoothness of the prediction surface
    - **boosting** and **random forest**: two major classification solutions that are built upon and greatly improve CART

Linear Support Vector Machines

Separating hyperplanes

- separating hyperplanes:
  - key idea: construct linear decision boundaries to explicitly separate the data into different classes as much as possible
  - code $Y = \pm 1$; the classification rule $= \text{sign}(f(\boldsymbol{x})) = \text{sign}(\beta_0 + \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x})$

- properties of hyperplane:
  - the affine set $L : \{\boldsymbol{x} | f(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x} = 0\}$
  - the normal vector of L: $\boldsymbol{\beta}^* = \boldsymbol{\beta}/\|\boldsymbol{\beta}\|$
  - for any $\boldsymbol{x}_0 \in L$, $\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x}_0 = -\beta_0$
  - the signed distance of any $\boldsymbol{x}$ to L: $\boldsymbol{\beta}^{*\mathsf{T}}(\boldsymbol{x} - \boldsymbol{x}_0) = f(\boldsymbol{x})/\|\boldsymbol{\beta}\|$
  - the width of margin between two hyperplanes $f(\boldsymbol{x}) = 1$ and $f(\boldsymbol{x}) = -1$: $2/\|\boldsymbol{\beta}\|$



Separating hyperplanes

- optimal separating hyperplane
  - when the two classes are linearly separable, there are more than one hyperplane that can separate the training points perfectly
  - idea: find a hyperplane that achieves biggest **margin** between the training points for 1 and -1

- optimization:

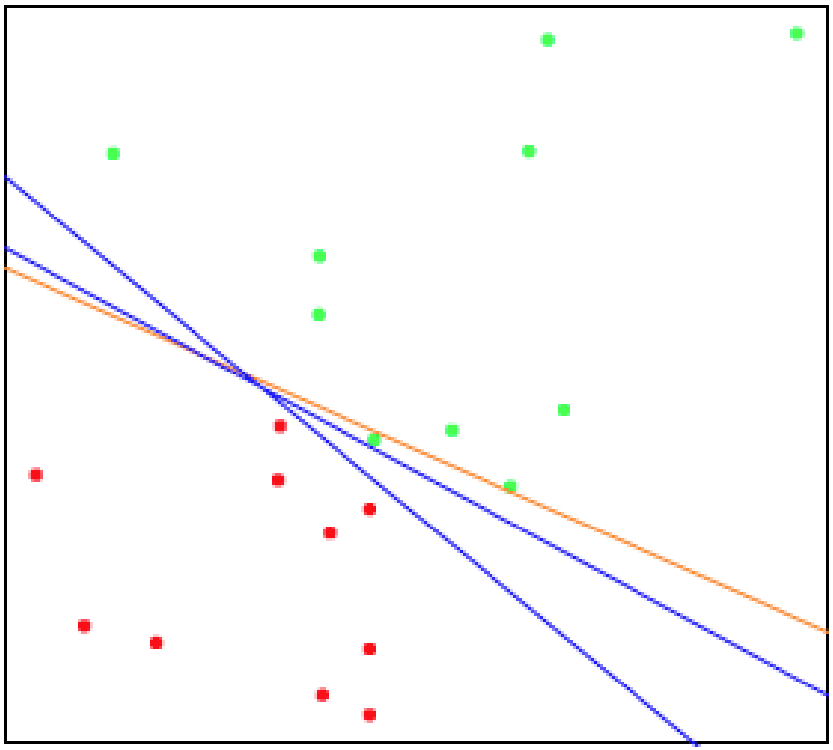
$$\max_{\beta_0,\beta,\|\beta\|=1} C \quad \text{subject to } \frac{1}{\|\beta\|} y_i(\boldsymbol{x}_i^\mathsf{T}\boldsymbol{\beta} + \beta_0) \geq C,\ i = 1, \ldots, n$$

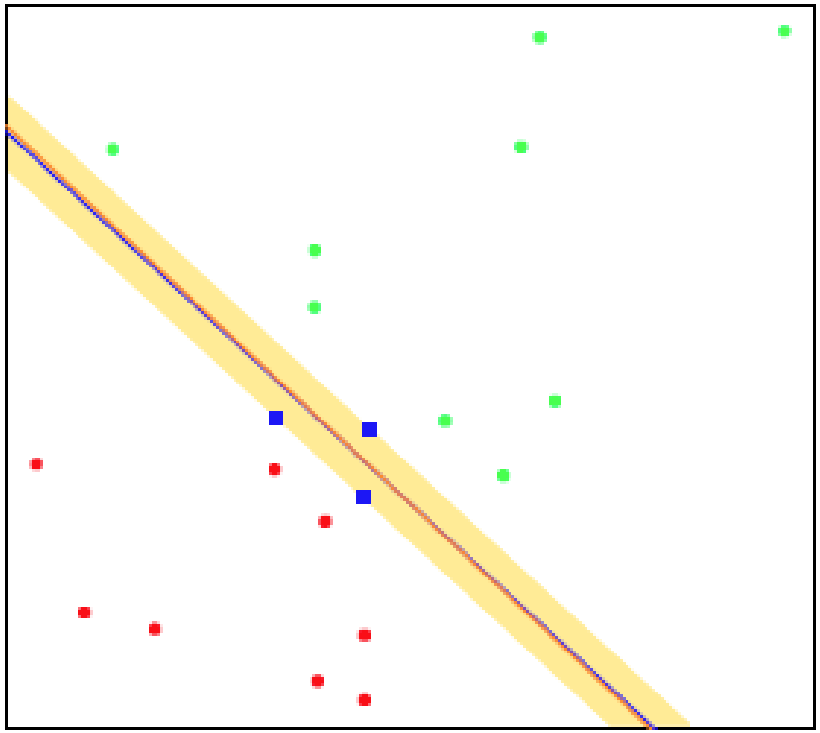
  i.e., all the points are at least a signed distance C from the decision boundary defined by $\boldsymbol{x}^\mathsf{T}\boldsymbol{\beta} + \beta_0 = 0$, and we seek the largest such C

- re-formulation: set $C\|\boldsymbol{\beta}\| = 1$

$$\min_{\beta_0,\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{subject to } y_i(\boldsymbol{x}_i^\mathsf{T}\boldsymbol{\beta} + \beta_0) \geq 1,\ i = 1, \ldots, n$$

Separating hyperplanes





Separating hyperplanes

- Lagrange primal objective function:

$$L_P = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \sum_{i=1}^{n} \alpha_i \{y_i(\boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta} + \beta_0) - 1\}.$$

- Wolfe dual objective function:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{x}_j, \quad \text{subject to } \alpha_i \geq 0$$

- Karush-Kuhn-Tucker Condition:

$$\begin{aligned}
0 &= \sum \alpha_i y_i \\
\boldsymbol{\beta} &= \sum \alpha_i y_i \boldsymbol{x}_i \\
\alpha_i &\geq 0, \quad \text{for} \quad i = 1, ..., n \\
0 &= \alpha_i [y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + \beta_0) - 1], \quad \text{for} \quad i = 1, ...n \\
1 &\leq y_i(\boldsymbol{x}_i^T \boldsymbol{\beta} + \beta_0), \quad \text{for} \quad i = 1, ...n
\end{aligned}$$

Separating hyperplanes

- solutions:
  - $\boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$;
  - $\alpha_i \{ y_i (\boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \beta_0) - 1 \} = 0, \forall i$.

- remarks:
  - $\boldsymbol{\beta}$ is a linear combination of $\boldsymbol{x}_i$'s with the signed coefficients α_i's
  - if $\alpha_i > 0$, then $y_i (\boldsymbol{x}_i^{\mathsf{T}} \beta + \beta_0) = 1$, so $\boldsymbol{x}_i$ is on the boundary of slab
  - if $y_i (\boldsymbol{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \beta_0) > 1$, then $\alpha_i = 0$, so $\boldsymbol{x}_i$ does not contribute to β
  - those $\boldsymbol{x}_i$'s with $\alpha_i > 0$ are called **support vectors**
  - the hyperplane depends only on those support vectors, and is thus more robust to model misspecification
  - identification of support vectors requires use of all data

Separating hyperplanes

- when the two classes are **not linearly separable**:

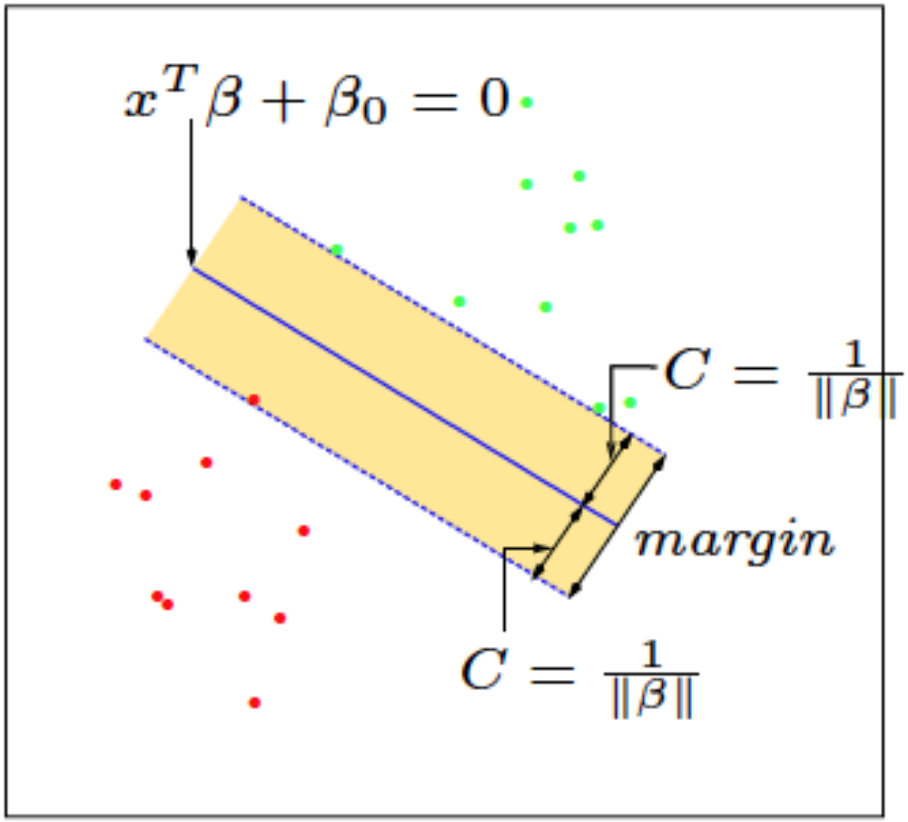
$$\min_{\beta_0,\beta} \frac{1}{2}\|\boldsymbol{\beta}\|^2 \quad \text{subject to } y_i(\boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta}+\beta_0) \geq 1-\xi_i, \xi_i \geq 0, \sum \xi_i \leq C^*$$

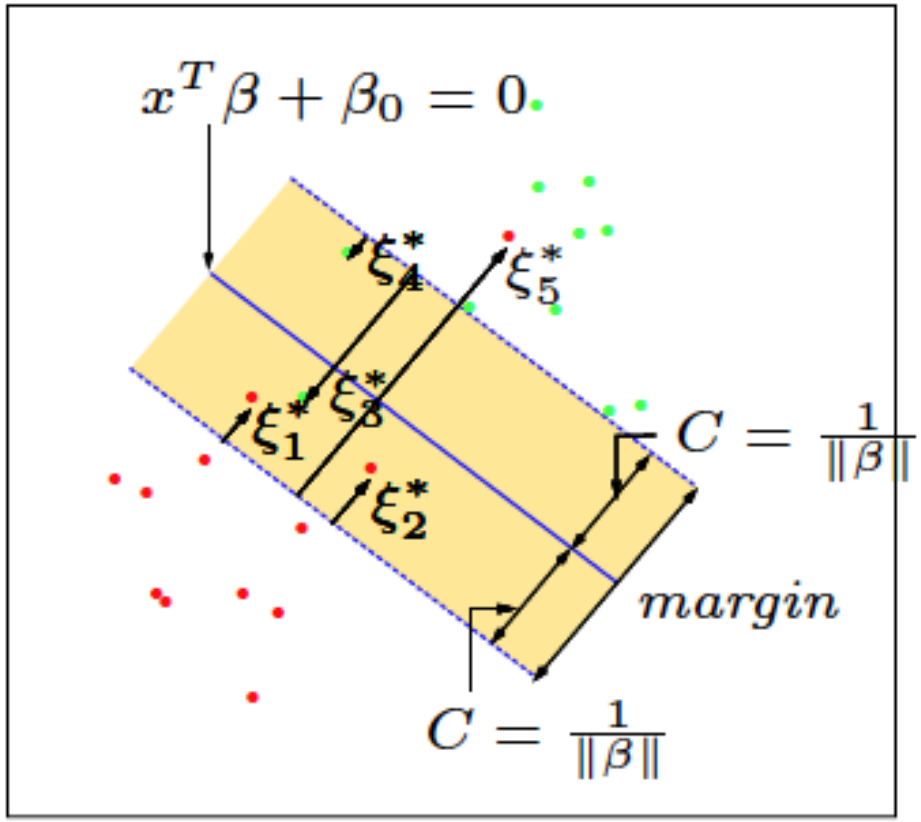
  i.e., we allow for some points to be on the wrong side of the margin, but we control the total distance of those points, which is measured by $\sum \xi_i$

- solutions:
  - $\boldsymbol{\beta} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$
  - $\alpha_i\{y_i(\boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{\beta}+\beta_0)-(1-\xi_i)\} = 0, \forall i$

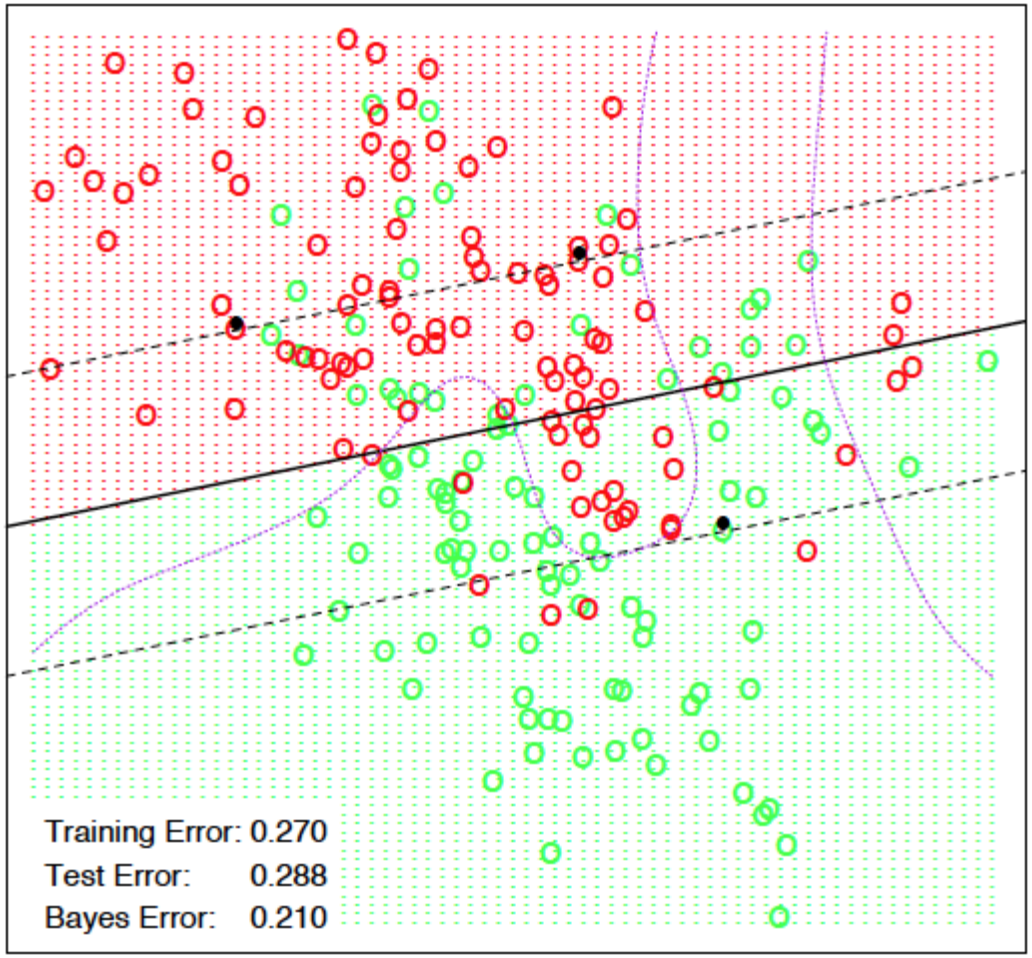
- **linear support vector machine**

Separating hyperplanes





Separating hyperplanes



Additional readings

- Cheng, G., and Shang, Z. (2015). Computational limits of divide-and-conquer method. *arXiv*.
- Cook, R.D. and Weisberg, S. (1999). *Applied Regression including Computing and Graphics*. Wiley.
- Cormode, G., and Duffield, N. (2014). Subsampling for Big Data. KDD2014.
- Grey, A. (2015). Big Data's small lie – the limitation of sampling and approximation in Big Data analysis.
- Taylor, J. *Stat 191: Introduction to Applied Statistics*, Lecture notes
- Diez, D.M., Barr, C.D., and Cetinkaya-Rundel, M. (2012). *OpenIntro Statistics*. http://www.openintro.org/stat/textbook.php. Chapter 7.
- Hastie, H., Tibshirani, R., and Friedman, J. (2001). *Elements of Statistical Learning*. Springer. Chapters, 4, 6, 9, 12.
- Lin, M., Lucas, H.C. Jr, Shmueli, G. (2013). Too big to fail: large samples and the p-value problem. *Information Systems Research*, 4, 906-917.
- Tian, L., Alizadeh, A.A., Gentles, A.J., and Tibshirani, R. (2014). A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association*, **109**, 1517-1532.