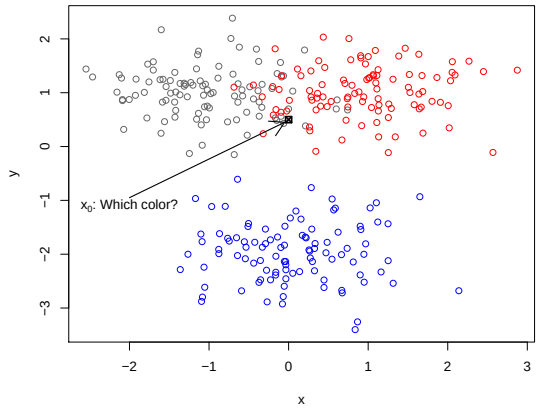
Review: Classification problem

Situation: Have observations $(x_1, y_1), \ldots, (x_n, y_n)$, where $y_i \in \{0, 1, \ldots, J-1\}$ gives a class. Have new observation x_0, want to predict the corresponding class y_0.



Review: Model

We have: $p_j = \mathrm{P}(Y = j)$, $f(x|y = j) = f_j(x)$

- $$\pi_j(x_0) = \mathrm{P}(Y_0 = j|x_0) = \frac{p_j f_j(x_0)}{\sum_{i=0}^{J-1} p_i f_i(x_0)}$$
- $$\mathrm{ECM}(i) = \mathrm{E}(c(i|Y)|x_0) = \frac{\sum_{j=0}^{J-1} c(i|j) p_j f_j(x_0)}{\sum_{i=0}^{J-1} p_i f_i(x_0)}$$
- $$\hat{y}_0 = \mathrm{argmin}_i \mathrm{ECM}(i) \overset{\text{0/1-loss}}{=} \mathrm{argmax}_i \{p_i f_i(x_0)\}$$

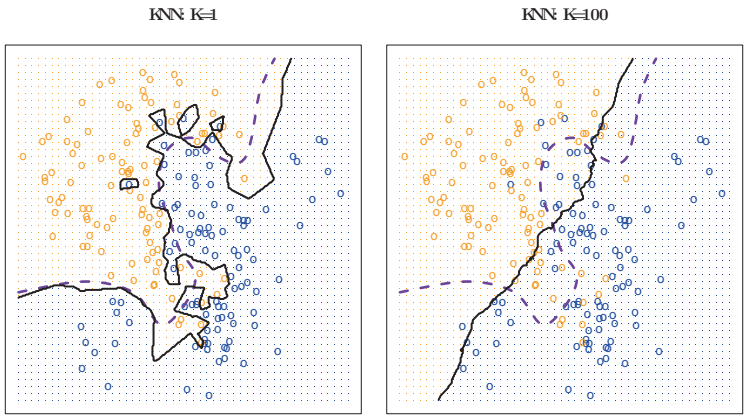
$\Rightarrow$

$$x|y = j \sim \mathcal{N}(\mu_j, \mathbf{\Sigma}_j) \begin{cases} \text{LDA} & \mathbf{\Sigma}_0 = \ldots = \mathbf{\Sigma}_{J-1} \\ \text{QDA} & \text{different } \mathbf{\Sigma}_j \end{cases}$$

Review

We have also discussed:

- k-nearest neighbour algorithm with tuning parameter k.



- Evaluation of classification rules: Today

Cross-validation

Consider a classification problem:

Have observed $(x_1, y_1), \ldots, (x_n, y_n) \leftarrow$ training data. Have one (or more) classification rule(s):

$$\hat{y}(x_0; (x_1, y_1), \ldots, (x_n, y_n))$$

How can we evaluate how good the rule is? Alternatively, how can we decide which rule is the best?

Misclassification rate

It is reasonable to focus on

- the misclassification rate

$$P(y_0 \neq \hat{y}(x_0; (x_1, y_1), \ldots, (x_n, y_n)))$$

, or

- expected cost (from misclassification)

$$E[c(\hat{y}(x; (x_1, y_1), \ldots, (x_n, y_n))|y)]$$

Apparent error rate

The apparent misclassification rate becomes

$$\frac{1}{n}\sum_{i=1}^{n} 1(y_i \neq \hat{y}(x_i; (x_1, y_1), \ldots, (x_n, y_n))$$

This estimate becomes clearly too optimistic because we use the same data to "train" the classifier and to estimate the misclassification rate.

We have to take into account:

- the assumed (parametric) model may be wrong.
- uncertainty in the parameter estimates
- inherent randomness

If we have a lot of training data ...

... the effect of parameter uncertainty is negligible and we can do the following:

1. divide the (training) data in two parts: training and test set
2. establish classifier from training set data
3. do classification for data in test data set, and estimate misclassification rate by the fraction of misclassification in test set.

Note: If we do not have so many training data this procedure will overestimate the misclassification rate, i.e. too pessimistic.

Idea k-fold cross validation

- Cross-validation can be use to estimate the misclassification rate of a statistical classification method.
- k-fold cross-validation involves randomly dividing the set of observations into k groups, or folds, $A_1, \ldots, A_k$ of approximately equal size.
- For the j-th fold (test set), we fit the model to the other $k-1$ folds (training set) of the data, and count the number of misclassifications of the fitted model when predicting the j-th part of the data.
- We do this for $j = 1, 2, \ldots, k$ and combine the k estimates
- Leave-one-out cross validation is a special case.

Leave-one-out cross validation (CV)

Let $\hat{y}(x) = \hat{y}(x; (x_1, y_1), \ldots, (x_n, y_n))$ denote our classifier based on all training data. Let

$$\hat{y}_{-i}(x) = \hat{y}(x; (x_1, y_1), \ldots, (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), \ldots, (x_n, y_n))$$

be our classifier based on all training data except (x_i, y_i).

Estimate the misclassification rate by:

$$\frac{1}{n} \sum_{i=1}^{n} 1(y_i \neq \hat{y}_{-i}(x_i))$$

Leave-one-out CV is computationally expensive. A cheaper variant is K-fold CV

K-fold CV

Divide at random training data into K sets $A_1, \ldots, A_K$ of equal size (or as close as possible). Let

$$\hat{y}_{-A_k}(x) = \hat{y}(x; (x_i, y_i), i \in \bigcup_{j \neq k} A_j)$$

and estimate the misclassification rate by

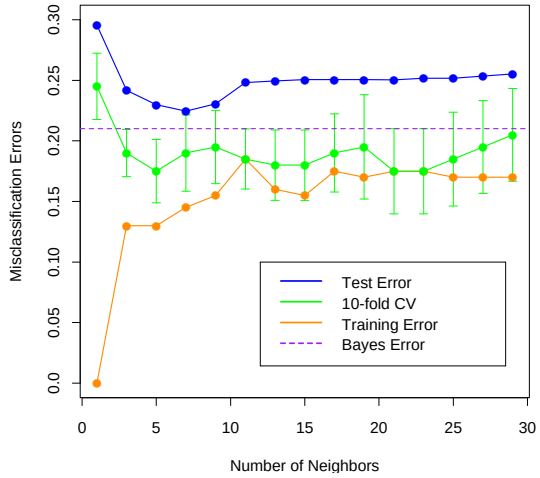
$$\frac{1}{n} \sum_{k=1}^{K} \left[\sum_{i \in A_k} 1(y_i \neq \hat{y}_{-A_k}(x_i)) \right].$$

Often, $K = 5$ or $K = 10$ is used.

Note: The tuning parameter k in the knn-classifier can be chosen using CV.

Show animation in R: `cv.ani` in animation package.

Misclassification as function of k



Hastie, Tibshirani, Friedman, "The elements of statistical learning", 2nd ed., p. 467

Performance of cross-validation

What are the advantages and disadvantages of k-fold cross-validation relative to leave-one-out cross-validation when $k < n$?

See Problem 4b, exam 2014.

Bootstrap

http://tradingconsequences.blogs.edina.ac.uk/files/2013/10/Dr_Martens_black_old.jpg

Bootstrap Bill Turner

"Bootstrap" Bill Turner from Pirates of the Caribbean.

> *. . . Barbossa tied Bill to a cannon by his bootstraps and sent him to the bottom of the sea.*

http://kidstvmovies.about.com/od/piratesofthecaribbean3/ig/Pirates-At-World-s-End/-Bootstrap-Bill.htm

. . . pull oneself up by one's bootstraps

> *To begin an enterprise or recover from a setback without any outside help; to succeed only on one's own effort or abilities.*
>
> Wiktionary

The term is sometimes attributed to Rudolf Erich Raspe's story "The Surprising Adventures of Baron Munchausen", where the main character pulls himself (and his horse) out of a swamp by his hair

http://redstateeclectic.typepad.com/redstate_commentary/2010/11/sustainability-isnt-sustainable.html

Bootstrapping in statistics

Bootstrap is a computer-based technique for doing statistical inference (usually with a minimum of assumptions). It is not Bayesian.

See blackboard for rough idea

Show animation in R: `boot.iid` in animation package.

Bootstrap principle

Assume we have iid observations from an (unknown) distribution F:

$$F \to (x_1, \ldots, x_n)$$

The empirical distribution function $\hat{F}$ is the CDF that puts mass $1/n$ at each data point x_i:

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} 1(x_i \leq x)$$

where $1(\cdot)$ denotes the indicator function.

Bootstrap principle

Let θ be an interesting feature of F, $\theta = T(F)$.

For example:

$$\theta = \mathrm{E}(X) = \int x f(x) dx$$

$$\theta = \mathrm{Var}(X) = \int (x - \mathrm{E}(X))^2 f(x) dx$$

The plug-in estimator for θ is defined by:

$$\hat{\theta} = T(\hat{F})$$

The plug-in principle is quite good, if the only information about F, comes from the sample x.

Examples

Thus

$$\theta = \mathrm{E}(X) \Rightarrow \hat{\theta} = \mathrm{E}_{\hat{F}}(X) = \sum_{i=1}^{n} x_i \frac{1}{n} = \bar{x}$$

$$\theta = \mathrm{Var}(X) \Rightarrow \hat{\theta} = \mathrm{Var}_{\hat{F}}(X) = \mathrm{E}_{\hat{F}}[(X - \mu_{\hat{F}})^2]$$

$$= \sum_{i=1}^{n} (x_i - \mu_{\hat{F}})^2 \frac{1}{n} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

$$\theta = \mathrm{SD}(X) \Rightarrow \hat{\theta} = \mathrm{SD}_{\hat{F}}(X) = \sqrt{\mathrm{Var}_{\hat{F}}(X)}$$

$$= \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Setting

Assume we have :

$$F \to (x_1, \ldots, x_n)$$

Thus $\hat{F}$ gives mass $\frac{1}{n}$ to each observed value.

A bootstrap sample is defined to be a random sample of size n from $\hat{F}$, say $x^\star = (x_1^\star, \ldots, x_n^\star)$

$$\hat{F} \to (x_1^\star, \ldots, x_n^\star)$$

Simple illustration

Suppose $n = 3$ univariate data points, namely

$$\{x_1, x_2, x_3\} = \{1, 2, 6\}$$

are observed as an iid sample from F that has mean θ . At each observed data value, $\hat{F}$ places mass $1/3$. Suppose the estimator to be bootstrapped is the sample mean $\hat{\theta}$.

There are $3^3 = 27$ possible outcomes for $\mathcal{X}^\star = \{X_1^\star, X_2^\star, X_3^\star\}$.

Simple illustration (II)

$\mathcal{X}^\star$	$\hat{\theta}^\star$	$P^\star(\hat{\theta}^\star)$	Observed frequency
1 1 1	3/3	1/27	36/1000
1 1 2	4/3	3/27	101/1000
1 2 2	5/3	3/27	123/1000
2 2 2	6/3	1/27	25/1000
1 1 6	8/3	3/27	104/1000
1 2 6	9/3	6/27	227/1000
2 2 6	10/3	3/27	131/1000
1 6 6	13/3	3/27	111/1000
2 6 6	14/3	3/27	102/1000
6 6 6	18/3	1/27	40/1000

Bootstrap estimate for standard error

- Parameter of interest: $\theta = T(F)$
- Our estimator for θ: $\hat{\theta} = s(x)$
- Want (to estimate) $\mathrm{SD}_F(\hat{\theta})$.

A bootstrap replication of $\hat{\theta}$ is

$$\hat{\theta}^\star = s(x^\star)$$

Use plug-in principle to estimate $\mathrm{SD}_F(\hat{\theta})$. The bootstrap estimate of the standard error of $\hat{\theta} = s(x)$ is $\mathrm{SD}_{\hat{F}}(\hat{\theta}^\star)$. This is called the ideal bootstrap estimate of standard error of $\hat{\theta}$.

Note: Except for very small n, $\mathrm{SD}_{\hat{F}}(\hat{\theta}^\star)$ cannot be computed.
(Number of possible bootstrap sample: n^n.)

Computational way of obtaining a good estimate

We can estimate $\mathrm{SD}_{\hat{F}}(\hat{\theta}^\star)$ by simulation:

1. Generate B bootstrap samples $x^{1\star}, \ldots, x^{B\star}$.
2. Evaluate the corresponding parameter estimates
$$\hat{\theta}^\star(b) = s(x^{b\star}), \quad b = 1, 2, \ldots, B$$
3. Estimate $\mathrm{SD}_{\hat{F}}(\hat{\theta}^\star)$ by
$$\widehat{\mathrm{SE}}_B = \sqrt{\frac{\sum_{b=1}^B (\hat{\theta}^\star(b) - \hat{\theta}^\star(\cdot))^2}{B-1}}$$
where
$$\hat{\theta}^\star(\cdot) = \frac{1}{B}\sum_{b=1}^B \hat{\theta}^\star(b)$$

Note
$$\lim_{B\to\infty} \widehat{\mathrm{SE}}_B = \widehat{\mathrm{SE}}_\infty = \widehat{\mathrm{SD}}_{\hat{F}}(\hat{\theta}^\star)$$

Example

Setting

$$\theta = \mathrm{E}(X)$$

$$\hat{\theta} = s(x) = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x}$$

$$\hat{\theta}^\star = s(x^\star) = \frac{1}{n}\sum_{i=1}^{n} x_i^\star = \bar{x^\star}$$

Here, the ideal bootstrap estimate exists

see blackboard

How large do we need B?

Intuitively we understand that the $\widehat{\mathrm{SE}}_B$ has larger standard deviation than $\widehat{\mathrm{SE}}_\infty$.

Theory, not to be discussed here, gives the following rules of thumb:

1. Even a small B is informative, say $B = 25$ or $B = 50$ is often enough to get a good estimate of $\mathrm{SE}_F(\hat{\theta})$.
2. Very seldomly more than $B = 200$ is necessary to estimate $\mathrm{SE}_F(\hat{\theta})$.

The parametric bootstrap

When data are modeled to originate from a parametric distribution, so

$$X_1, \ldots, X_n \overset{\text{iid}}{\sim} F(x, \xi),$$

another estimate of F may be employed.

Suppose that the observed data are used to estimate ξ by $\hat{\xi}$. Then each parametric bootstrap pseudo-dataset $\mathcal{X}^\star$ can be generated by drawing $X_1^\star, \ldots, X_n^\star \overset{\text{iid}}{\sim} F(x, \hat{\xi}) = \hat{F}_{\text{par}}$.

Again ...

... we can/must estimate $\mathrm{SD}_{\hat{F}_{\text{par}}}(\hat{\theta}^\star)$ by simulation:

1. Generate B bootstrap samples $x^{1\star}, \ldots, x^{B\star}$, where
$$x^{b\star} = (x_1^{b\star}, \ldots, x_n^{b\star})$$
with $x_1^{b\star}, \ldots, x_n^{b\star} \overset{\text{iid}}{\sim} \hat{F}_{\text{par}}$.
2. Evaluate the corresponding parameter estimates
$$\hat{\theta}^\star(b) = s(x^{b\star}), \quad b = 1, 2, \ldots, B$$
3. Estimate $\mathrm{SD}_{\hat{F}_{\text{par}}}(\hat{\theta}^\star)$ by
$$\widehat{\mathrm{SE}}_B = \sqrt{\frac{\sum_{b=1}^{B}(\hat{\theta}^\star(b) - \hat{\theta}^\star(\cdot))^2}{B-1}}$$
where
$$\hat{\theta}^\star(\cdot) = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}^\star(b)$$

Bootstrapping regression

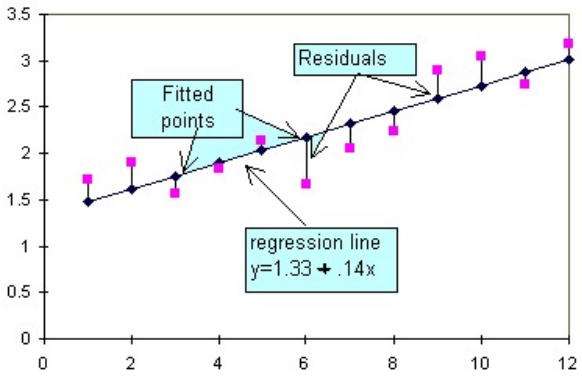
Consider the ordinary multiple regression model

$$Y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \quad \text{for } i = 1, \ldots, n,$$

where ϵ_i are iid mean zero random variables with constant variance.

- Naive: Bootstrapping by resampling from response variables to get distribution of $\hat{\boldsymbol{\beta}}^\star$. However $Y_i | \boldsymbol{x}_i$ are not iid!
- Correct: Bootstrap the residuals.

Review: Residuals



http://fsweb.bainbridge.edu/dbyrd/statistics/regression.htm

Bootstrap the residuals

1. Fit the regression model to the observed data and obtain the fitted responses $\hat{y}_i$ and residuals $\hat{\epsilon}_i$.
2. Sample a bootstrap set of residuals $\hat{\epsilon}_1^\star, \ldots, \hat{\epsilon}_n^\star$ from the set of fitted residuals completely at random and with replacement.
3. Generate a bootstrap set of pseudo responses

$$Y_i^\star = \hat{y}_i + \hat{\epsilon}_i^\star, \quad \text{for } i = 1, \ldots, n.$$

4. Regress $Y^\star$ on $\boldsymbol{x}$ to obtain a bootstrap estimate $\hat{\boldsymbol{\beta}}^\star$.

Repeat this process to get an empirical distribution of $\hat{\boldsymbol{\beta}}^\star$.

Bootstrapping residuals: Remarks

This approach is also used for autoregressive models, for example.

Note: Bootstrapping the residuals is reliant on

- The model provides an appropriate fit
- The residuals have a constant variance

Otherwise, a different scheme is recommended.

Comment: No need to bootstrap for linear regression model and least squares estimation, as analytical results are then available.